

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено  
Завідувач кафедри

О.В.Коваль  
(ініціали, прізвище)

\_\_\_\_\_  
(підпис)

“ \_\_\_\_ ” \_\_\_\_\_ 2019 р.

**ДИПЛОМНА РОБОТА**  
**на здобуття ступеня бакалавра**

з напрямку підготовки  
6.050101 “Комп’ютерні науки”

на тему: Інформаційна система ефективного менеджменту роботи закладів  
післядипломної освіти

Виконала: студентка 4 курсу, групи ТР-52

Басалик Дар’я Андріївна  
(прізвище, ім’я, по батькові)

\_\_\_\_\_  
(підпис)

Керівник ст.вик. Бандурка Олена Іванівна  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

\_\_\_\_\_  
(підпис)

Рецензент \_\_\_\_\_  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

\_\_\_\_\_  
(підпис)

Засвідчую, що у цій дипломній роботі немає  
запозичень з праць інших авторів без  
відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Київ – 2019

**Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший рівень

Напрямок підготовки 6.050101 “Комп’ютерні науки”

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ О.В. Коваль  
(підпис)

” \_\_\_\_ ” \_\_\_\_\_ 2019 р.

**ЗАВДАННЯ**

**на дипломну роботу студентці**

Басалик Дар’ї Андріївни  
(прізвище, ім’я, по батькові)

1. Тема роботи Інформаційна система ефективного менеджменту роботи закладів післядипломної освіти

керівник роботи ст.вик. Бандурка Олена Іванівна

(прізвище, ім’я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від ” \_\_\_\_ ” \_\_\_\_\_ 201\_\_ р.  
№ \_\_\_\_\_

2. Строк подання студентом роботи \_\_\_\_ 201\_\_ р.

3. Вихідні дані до роботи Excel-файл з таблицею розкладу та навантаження на викладачів

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) проаналізувати існуючі програмні рішення для ефективного менеджменту закладів освіти, описати програмну реалізацію програмного додатку, розробити програмне забезпечення, розробити інтерфейс користувача

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

5. Перелік ілюстраційного матеріалу (з точним зазначенням обов'язкових креслень) 1. Мета та завдання роботи 2. Огляд існуючих програмних рішень 3. Вимоги до модуля створення розкладу 4. Використані програмні засоби 5. Висновки

Дата видачі завдання "10" жовтня 2018 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Вивчення та аналіз задачі		
2.	Розробка архітектури та загальної структури системи		
3.	Розробка структур окремих підсистем		
4.	Підготовка матеріалів		
5.	Програмна реалізація системи		
6.	Захист програмного продукту	15.05.2019	
7.	Оформлення пояснювальної записки		
8.	Передзахист	28.05.2019	
9.	Захист		

Студентка

\_\_\_\_\_  
(підпис)

Басалик Д. А.

\_\_\_\_\_  
(прізвище та ініціали)

Керівник роботи

\_\_\_\_\_  
(підпис)

Бандурка О.І.

\_\_\_\_\_  
(прізвище та ініціали)

## **АНОТАЦІЯ**

Мета роботи — інформаційна система, яка складається з двох модулів: модулю менеджменту задач та модулю для побудови розкладу навчального закладу, що базується на базі даних. В модулі побудови розкладу враховано можливість додавати до бази нові дані про заклад освіти, перевіряти вхідні дані до блоку розкладу на коректність та формувати таблицю навантаження на викладачів. Кінцевий розклад та таблицю навантажень користувач зберігає до Excel-файлу. В модулі менеджменту задач переглядається поточний план робіт та підсвічуються записи в залежності від того, коли підходить їх кінцевий строк виконання.

Записка містить 58 сторінок, 36 рисунків та 17 посилань.

Ключові слова: C#, MICROSOFT SQL SERVER, БАЗА ДАНИХ, SQL, DATAGRIDVIEW, ТАБЛИЦІ, ЗАПИТ, РОЗКЛАД, МЕНЕДЖМЕНТ

## **ABSTRACT**

The purpose of the work is an information system, which consists of two modules: a task management module and a module for schedule building. The modulus for constructing the schedule includes the ability to add new data about the institution of education to the database, check the input data to the block of the schedule for correctness and to form a table of load for the teachers. The final timetable and the user's table of contents are saved to the Excel file. In the task management module the current plan of work is reviewed and the records are highlighted depending on when their deadline is appropriate.

The note contains 58 pages, 36 images and 17 references.

Keywords: C#, MICROSOFT SQL SERVER, DATABASE, SQL, DATAGRIDVIEW, TABLES, QUERY, SCHEDULE, MANAGEMENT

# ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	8
ВСТУП.....	9
1 ПОСТАНОВКА ЗАДАЧІ ЕФЕКТИВНОГО МЕНЕДЖМЕНТУ РОБОТИ ЗАКЛАДУ ПІСЛЯДИПЛОМНОЇ ОСВІТИ .....	11
2 АНАЛІЗ МЕТОДІВ ВИРІШЕННЯ ПРОБЛЕМИ ЕФЕКТИВНОГО МЕНЕДЖМЕНТУ РОБОТИ ЗАКЛАДІВ ОСВІТИ.....	13
2.1 Генетичний алгоритм.....	13
2.2 Аналіз існуючих програмних рішень.....	14
2.2.1 Програма “Ректор-ВУЗ” .....	14
2.2.2 Програма “Галактика Розклад навчальних занять” .....	15
2.2.3 Програми таск-менеджери .....	15
2.3 Особливості розкладу Національної медичної академії післядипломної освіти П.Л. Шупіка .....	15
2.4 Висновки до розділу 2 .....	16
3 ЗАСОБИ РОЗРОБКИ .....	18
3.1 Середовище розробки Visual Studio 2017 .....	18
3.2 Windows Forms .....	19
3.3 Бібліотека Microsoft.Office.Interop.Excel .....	20
3.4 Бібліотека System.Data.Odbc .....	20
3.5 Бібліотека System.Data.SqlClient .....	21
3.6 Microsoft SQL Server .....	21
3.7 Технологія ADO.NET .....	23
3.8 Висновки до розділу 3 .....	24
4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ .....	25
4.1 Діаграми прецедентів.....	25

4.2 Створення бази даних в Microsoft SQL Server .....	30
4.2.1 Створення бази даних для модуля формування розкладу .....	30
4.1.2 Створення бази даних для модуля менеджера задач.....	34
4.3 Підключення бази даних модуля формування розкладу до проекту Visual Studio та робота з таблицями .....	35
4.4 Підключення бази даних модуля менеджера задач до проекту Visual Studio та робота з таблицями .....	39
4.5 Висновки до розділу 4 .....	40
5 РОБОТА КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ.....	41
5.1 Системні вимоги.....	41
5.2 Робота з програмним модулем для складання розкладу.....	42
5.3 Робота з програмним модулем менеджменту задач .....	52
5.4 Висновки до розділу 5 .....	55
ВИСНОВКИ.....	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	57
ДОДАТОК А.....	59
ДОДАТОК Б.....	61
ДОДАТОК В .....	74
ДОДАТОК Г.....	83

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

СУБД — система управління базами даних;

БД — база даних;

SQL (англ. Select query language) — декларативна мова програмування для взаємодії користувача з базами даних;

MS (англ. Microsoft) — багатонаціональна корпорація комп'ютерних технологій.

UML (англ. Unified Modeling Language) — уніфікована мова моделювання

## ВСТУП

В сучасному світі перед керівниками навчальних закладів все частіше постає питання автоматизації певних процесів. Зокрема, автоматизація створення графіку навчального процесу та менеджменту задач, які повинні бути виконані співробітниками закладів [1]. Перевагами таких систем є економія часу співробітників, а також структурованість та організованість процесу виконання [2]. Неможливо забути про завдання або строк його здачі, тому що вони відображаються в програмі.

Раціональна організація навчального процесу є однією з найважливіших завдань ефективного функціонування вузу. Навчальний процес може бути описаний сукупністю взаємопов'язаних завдань, що вирішуються різними підрозділами навчального закладу. Об'єктом автоматизації в цьому випадку може бути викладацький склад, сукупність предметів, що викладаються або база кабінетів, де проводяться пари. Метою впровадження систем автоматизації є підвищення якості освіти.

Завдання побудови розкладу навчальних занять у ВНЗ вирішується в усіх університетах та інших навчальних закладах. Процес побудови розкладу є найбільш важливим в діяльності ВНЗ в цілому і для кожного факультету зокрема. Добре розроблений розклад може стати основою для реалізації науково-педагогічного потенціалу навчальних курсів і для кращого засвоєння матеріалу студентами. Завдання розподілу навчальної роботи між співробітниками кафедри, розробки навчального розкладу в значній мірі визначають ефективність організації освітнього процесу. Правильно і точно складений розклад забезпечує рівномірне завантаження студентських груп і професорсько-викладацького складу [3]. Від вдало складеного розкладу залежить ефективність роботи викладачів, засвоєння навчального матеріалу студентами, раціональне використання інтелектуальної та матеріальної баз навчального закладу.



Дослідження проблем менеджменту роботи на кафедрі вірусології Національної медичної академії післядипломної освіти П.Л. Шупіка сприяло розробці програмного продукту, який вирішує проблеми автоматизації менеджменту кафедри.

Звіт з переддипломної практики містить 5 розділів.

У першому розділі описується постановка задачі інформаційної системи ефективного менеджменту роботи закладів післядипломної освіти.

У другому розділі описуються підходи до вирішення проблеми ефективного менеджменту роботи закладів післядипломної освіти.

У третьому розділі описуються засоби розробки системи.

У четвертому розділі описується поетапна програмна реалізація додатку.

У п'ятому розділі описано роботу користувача з програмною системою.

# **1 ПОСТАНОВКА ЗАДАЧІ ЕФЕКТИВНОГО МЕНЕДЖМЕНТУ РОБОТИ ЗАКЛАДУ ПІСЛЯДИПЛОМНОЇ ОСВІТИ**

На основі відомостей та теоретичного матеріалу, який був наданий керівником дипломної роботи та керівником практики, розробити інформаційну систему ефективного менеджменту роботи закладів післядипломної освіти. Система має складатися з двох модулів:

- модуль менеджменту задач;
- модуль для побудови розкладу навчального закладу.

В модулі побудови розкладу необхідно було врахувати можливість додавати до бази нові предмети, спеціалізації, кабінети, групи та викладачів. При додаванні до розкладу нового блоку повинні проводитися певні перевірки на коректність та правильність введених даних. Після формування загального розкладу користувач повинен мати змогу сформувати навантаження для викладачів за певний період часу та розклад для конкретної спеціалізації. Усі сформовані таблиці користувач має змогу зберегти до Excel-файлів. Кінцевий документ з розкладом має містити наступні колонки:

- дата;
- пара;
- код заняття;
- тематика заняття;
- викладач;
- група;
- аудиторія.

В модулі моніторингу виконання задач необхідно надати користувачу можливість заносити основний план робіт до бази та зчитувати з Excel-файлу додаткові плани. Модуль повинен бути розроблений конкретно під структуру

документів Національної медичної академії післядипломної освіти П.Л. Шупіка, а тому містити такі колонки:

- рішення Вченої ради;
- дата та номер протоколу Вченої ради;
- термін виконання;
- відповідальні особи за рішення;
- етап виконання;
- виконано.

Для роботи з даними було вирішено створити дві бази даних за допомогою системи управління реляційними базами даних Microsoft SQL Server. Безпосередньо інформаційна система написана на мові програмування С#. Графічний інтерфейс розроблено за допомогою інтерфейсу програмування додатків Windows Forms.

Метою розробки інформаційної системи ефективного менеджменту роботи закладів післядипломної освіти є програмний продукт, який вирішить проблему зірваних строків виконання, помилок, які можливі при складанні розкладу вручну, а також збереже час співробітників закладу, завдяки автоматизації цих кропітких задач.

## **2 АНАЛІЗ МЕТОДІВ ВИРІШЕННЯ ПРОБЛЕМИ ЕФЕКТИВНОГО МЕНЕДЖМЕНТУ РОБОТИ ЗАКЛАДІВ ПІСЛЯДИПЛОМНОЇ ОСВІТИ**

На сьогоднішній день проблема розробки розкладу навчання залишається актуальною. Вирішенням цієї задачі займається багато математиків та програмістів, проте досі не було розроблено додатку, який був би актуальним для усіх вишів. Існує багато математичних постановок задачі, проте усі вони різняться вхідними параметрами, а отже не є універсальними. Існуючі додатки представлені лише на комерційних ринках. Проте через різні умови навчання, обмеження та вхідні дані для кожного окремого університету багатьом вишам немає сенсу їх купувати. Ці додатки не вирішують проблеми побудови розкладу кожного конкретного вищого навчального закладу.

Для вирішення проблеми створення розкладу вищого навчального закладу пропонують використовувати: класичні методи (теорія графів), мета евристичні методи (еволюційні та генетичні алгоритми, табу-пошук, метод мурашиних колоній, максимальна мурашина система, оптимізація роєм частинок), мультиагентні системи, методи рішення по прецедентам [4].

### **2.1 Генетичний алгоритм**

На сьогоднішній день найбільш популярним серед дослідників вважається генетичний алгоритм. Генетичний алгоритм — потужний засіб, який може застосовуватися для вирішення дуже складних задач, де неможливо застосувати спеціальні методи. Принцип роботи алгоритму наступний:

1. Формування початкової популяції.
2. Селекція особин.

3. Схрещування особин випадковими значеннями функції придатності.
4. Операція мутації над нащадками.
5. Перевірка значення гена «конфліктів» та його ліквідація у разі необхідності.
6. Формування нової популяції.
7. Перевірка критерію зупинки алгоритму.
8. Вибір найкращої особини.

В випадку задачі побудови розкладу популяція складається з  $N$  кількості особин, де кожна особина — окремий варіант розкладу. Всі особини складаються з трьох хромосом, які представлені аудиторіями, часом та блоком. Блок включає в себе назву предмета та викладача, який цей предмет читає [5].

Проте значним недоліком генетичного алгоритму є час його роботи, на знаходження лише одного варіанту розкладу може піти невизначена кількість часу.

## **2.2 Аналіз існуючих програмних рішень**

### **2.2.1 Програма “Ректор-ВУЗ”**

Програма складається з чотирьох розділів:

- “Списки”, в якому користувач вводить дані, які стосуються кількості кафедр, груп, дисциплін, аудиторій, типів занять та викладачів;
- “Навантаження”, в якому вводяться учбові плани по спеціальностям, навантаження на викладачів та звіти по ньому;
- “Розклад”, в якому безпосередньо складається розклад навчання;
- “Заміни”, в якому можна замінювати викладачів.

Розклад можна складати в автоматичному, ручному або комбінованому режимі. Готовий розклад можна зберегти в форматах Microsoft Word, Excel та HTML.

Недоліками програми є те, що розклад, який формується автоматично, потрібно ще довго переробляти вручну та те, що цей додаток можна лише купити, у відкритому доступі його немає.

### **2.2.2 Програма “Галактика Розклад навчальних занять”**

Процес формування розкладу в цій системі виконується в декілька етапів.

Спочатку користувач має заповнити усі дані, стосовно системи та правил навчання. Потім заносяться усі обмеження, яких необхідно дотриматися при складанні розкладу, причому вони розбиваються на обов’язкові та бажані. Потім можна приступати до етапу формування розкладу — автоматично або вручну. Після того, як розклад сформовано, до нього можна ввести деякі зміни вручну.

Недоліками цієї програми є те, що вона платна та розклад, який формується автоматично, за відгуками користувачів, дуже важко впроваджувати в життя і його необхідно все одно переробляти.

### **2.2.3 Програми таск-менеджери**

Програми таск-менеджери, які знаходяться у вільному доступі, не задовольняють керівників Національної медичної академії післядипломної освіти П.Л. Шупіка, тому що таблиці додатків не мають необхідних полів, які є важливими для академії. Також ці програми не мають функціоналу для зчитування даних з Excel-файлів, а отже користувачам необхідно було б кожен раз витратити час, щоб перенести дані.

## **2.3 Особливості розкладу Національної медичної академії післядипломної освіти П.Л. Шупіка**

В процесі дослідження роботи кафедри вірусології виявилось, що розклад у закладах післядипломної освіти — це не той розклад, який ми звикли бачити у КПІ або інших університетах.

Головною відмінністю, яка унеможливорює використання генетичного алгоритму є те, що розклад складається на кожен день різний. Навчання проходить

не семестрами, а циклами і не для курсів, а для спеціалізацій. У той момент, коли навчання проходить, наприклад, у спеціалізації «Вірусологія» до розкладу може приєднатися інша спеціалізація. Таким чином, коли вірусологи будуть закінчувати цикл, інша спеціалізація буде лише в середині навчання. Заздалегідь коли яка спеціалізація починає навчання сказати не можна.

Також відрізняється те, що будь-який кабінет підходить під будь-який тип пари (лекція, семінар або практика). Будь-який викладач може прочитати будь-яку тему. Спеціалісти, які отримують післядипломну освіту, не вивчають предмети (як у нас, наприклад, вища математика) — вони вивчають теми. Зазвичай одна тема триває одну пару, а отже розклад на кожен день є унікальним.

Обмеження, які вводяться на розклад:

- дві групи не можуть бути в одній аудиторії;
- два викладача не може викладати в одній аудиторії;
- одна група не може бути одночасно в двох аудиторіях;
- один викладач не може викладати одночасно в двох аудиторіях;
- один викладач не може викладати одночасно в двох групах;
- два викладача не може викладати одночасно в одній групі.

Але, якщо тип заняття — “Лекція”, тоді декілька груп може бути в одній аудиторії та один викладач може одночасно читати більш, ніж в двох групах.

Зважаючи на вищеописані особливості розкладу, було прийнято рішення створити систему, в якій розклад все ж заповнюється вручну, проте враховуються усі можливі помилки і деякі етапи заповнення автоматизовані.

## **2.4 Висновки до розділу 2**

В розділі номер 2 було розглянуто основні методи та програми, що допомагають вирішувати проблеми менеджменту закладів післядипломної освіти та особливості розкладу конкретного навчального закладу.

Внаслідок аналізу було виявлено, що досі немає універсального рішення для задачі складання розкладу навчання.

Алгоритми, які зараз застосовуються для вирішення цієї проблеми занадто громіздкі та важкі для програмування. Вони можуть шукати оптимальні розв'язки протягом довгого часу і немає гарантій, що знайдене рішення буде дійсно реальним для впровадження в життя.

Наявні на ринку програми не можуть повністю автоматизувати цей процес, так як використовують ці або навіть більш застарілі алгоритми. Програмні додатки добре працюють у режимі створення розкладу вручну, але не в автоматизованому. Розклади, створені в автоматизованому режимі, необхідно ще певний час дороблювати вручну, що зводить усі процеси автоматизації нанівець.



## **3 ЗАСОБИ РОЗРОБКИ**

Щоб правильно реалізувати завдання, необхідно обрати найбільш оптимальні засоби розробки (мову програмування та технології). Було проведено аналіз та обрано методи програмної реалізації для найкращого виконання завдання.

Середовищем розробки програмного продукту було обрано Microsoft Visual Studio 2017.

Графічний інтерфейс розроблявся за допомогою інтерфейса програмування додатків Windows Forms.

Код написано на об'єктно-орієнтовній мові програмування C#.

Для зчитування вхідної інформації по планам робіт та зберігання розкладу у Excel-файл була використана бібліотека Microsoft.Office.Interop.Excel.

Для створення бази даних було використано Microsoft SQL Server 2014.

Для зв'язку та роботи з базою даних та SQL-запитами було використано бібліотеки System.Data.Odbc та System.Data.SqlClient.

### **3.1 Середовище розробки Visual Studio 2017**

Visual Studio — це повністю інтегроване середовище розробки. Воно спроектоване таким чином, щоб робити процес написання коду, його налагодження та компіляції в збірку для поставки кінцевим споживачам якомога простішим [6]. Основні можливості Visual Studio 2017:

1. Текстовий редактор, в якому можна писати програми на різних мовах програмування, зокрема C#. Редактор тексту автоматично проставляє табуляцію та потрібні відступи, що полегшує читання коду, закриває дужки, формує блоки коду, виділяє ключові слова різним кольором.

2. Візуальний редактор форм, який автоматично додає до основного коду програми код елементів управління.

3. Інтегрований відладник, який дозволяє ставити точки початку та кінця відладки не виходячи із середовища розробки.

4. Довідкова система MSDN, до якої можна звертатися із середовища розробки. Досить виділити незрозуміле слово або помилку в коді, натиснути F1 і система покаже потрібні розділи довідки.

5. Доступ до інших програм, зокрема SQL Server, який допомагає переглядати дані в таблицях та налагоджувати зв'язок з базами даних.

6. В складі Visual Studio є цілий набір типових проектів, з яких розробник може обрати саме той, який йому зараз необхідний. Кожен раз, коли створюється новий проект, Visual Studio автоматично створює “скелет” майбутнього додатку, причому цей код можна одразу скомпілювати та запустити [7].

7. Доступ та швидке встановлення різних додаткових пакетів та розширень для MS Visual Studio, які полегшують роботу з кодом та створення проектів.

## **3.2 Windows Forms**

Для створення графічних інтерфейсів за допомогою платформи .NET застосовуються різні технології – Windows Forms, WPF, додатки для магазину Windows Store. Проте найбільш простою та зручною платформою досі залишається Windows Forms або форми [8].

Windows Forms використовуються, зокрема для розробки десктопних додатків. Їх перевагою є те, що вони стабільно працюють без доступу в Інтернет та доступуються до ресурсів на комп'ютері більш безпечним шляхом.

Форма — це візуальна поверхня, на якій відображається інформація для користувача. Коли користувач намагається зробити щось з елементом форми, додаток реагує на цю дію та генерує подію, яка обробляється за допомогою коду, який прописаний для кожного елемента.

Перевагою Windows Forms при роботі з базами даних є те, що вони підтримують архітектуру прив'язки даних та мають клас DataGridView, який полегшує відображення даних.

Клас `DataGridView` дає користувачеві настроювану таблицю. для відображення даних. Він дозволяє налаштувати комірки, рядки, стовпчики за допомогою таких властивостей як `DefaultCellStyle`, `ColumnHeadersDefaultCellStyle`, `CellBorderStyle` та `GridColor` [9].

Класи `Windows Forms` знаходяться в просторі імен `System.Windows.Forms`.

### 3.3 Бібліотека `Microsoft.Office.Interop.Excel`

Бібліотека `Microsoft.Office.Interop.Excel` у середовищі `Visual Studio` полегшує роботу з `Excel`-файлами.

Головним є інтерфейс `_Application`. Він містить налаштування та параметри, наприклад багато опцій у вікні «Параметри», а також властивості, які повертають об'єкти вищих рівнів, наприклад `ActivCell`, `ActiveSheet` та інші.

Інтерфейс `_Workbook`. Об'єкт `Workbook` — це член колекції `Workbooks`. Колекція `Workbooks` містить усі об'єкти `Workbook`, які в даний момент відкриті у `Microsoft Excel`. Цей об'єкт представляє сторінку документа `Excel` та увесь її зміст.

Клас `_Worksheet` — об'єкт колекції `Sheets`. Колекція `Sheets` містить усі листи в робочій книзі.

### 3.4 Бібліотека `System.Data.Odbc`

Провайдер даних `.NET Framework` для технологій `ODBC` описує набір класів, які використовуються для доступу до джерела даних `ODBC`. Основні класи — `OdbcCommand`, `OdbcConnection`, `OdbcParameter` та інші.

Клас `OdbcCommand` представляє оператор `SQL` або збережену процедуру для виконання певної дії над джерелом класу.

Клас `OdbcConnection` представляє відкрите підключення до джерела даних.

Клас `OdbcParameter` представляє параметр у `OdbcCommand`. Не може бути успадкований[10].

### 3.5 Бібліотека System.Data.SqlClient

Постачальник даних SqlConnection спеціально призначений та розроблений для роботи з SQL Server. Практично усі дії виконуються завдяки керованому коду.

System.Data.SqlClient — простір імен у провайдері SqlConnection даних для SQL Server. Основні класи SqlCommand, SqlConnection, SqlDataAdapter, SqlDataReader.

Клас SqlCommand представляє оператор Transact-SQL або збережену процедуру для виконання запитів над базою даних SQL Server.

Клас SqlConnection представляє підключення до бази даних SQL Server.

Клас SqlDataAdapter представляє набір команд та підключення до бази даних, які використовуються для заповнення та оновлення набору даних.

Клас SqlDataReader допомагає зчитувати потік рядків з бази даних для подальшої маніпуляції над ними у коді програми.

### 3.6 Microsoft SQL Server

Система управління базами даних SQL Server — це повнофункціональна база даних з основною мовою запитів Transact-SQL. Проте SQL Server підтримує і ANSI SQL, стандартну мову SQL. Дана СУБД вирішує питання управління даними для проектів різних масштабів: від невеликих додатків до високонавантажених систем[11].

Центральним аспектом в MS SQL Server, як і в будь-якій СУБД, є база даних. База даних являє сховище даних, організованих певним способом. Нерідко фізично база даних представляє файл на жорсткому диску, хоча таке відповідність необов'язково.

Для зберігання і адміністрування баз даних застосовуються системи управління базами даних (database management system) або СУБД (DBMS). І якраз MS SQL Server є однією з такою СУБД.

SQL Server пропонується в декількох виданнях з різним набором функцій і опціями ціноутворення для задоволення різноманітних потреб користувачів, включаючи такі:

1. Enterprise — призначено для великих підприємств зі складними вимогами до даних, сховищами даних і базами даних з підтримкою Web. Має всі можливості SQL Server і є найдорожчим.

2. Standard — націлений на малі та середні організації. Також підтримує електронну комерцію та зберігання даних.

3. Workgroup — для невеликих організацій. Немає обмежень розміру, може використовуватися як база даних для малих веб-серверів або філій.

4. Express — безкоштовно для розповсюдження. Має найменшу кількість функцій і обмежує розмір бази даних і користувачів. Може використовуватися як заміна бази даних Access.

Для реалізації дипломного проекту було обрано видання Express. Його характеристики наступні:

— максимальна обчислювальна потужність, що використовується однією інстанцією (SQL Server Database Engine): менше, ніж 4 ядра;

— максимальна обчислювана потужність, що використовується однією інстанцією (Analysis Services, Reporting Services): менше, ніж 4 ядра;

— максимальна використовувана пам'ять (за екземпляр SQL Server Database Engine): 1 ГБ;

— максимальна кількість використовуваної пам'яті (для екземпляра служб Analysis Services): недоступна;

— максимальна кількість використовуваної пам'яті (на примірник служб Reporting Services): недоступна;

— максимальний розмір реляційної бази даних: 10 ГБ [12].

Інші доступні функції та сервіси — це відновлення бази даних, серверна підтримка, затримка довговічності, визначення користувачем ролей, заповнення баз даних (таблиць) даними, інтеграція с Microsoft Visual Studio, підтримка типів даних Date та Time, SQL Server Import and Export Wizard та інші.

### 3.7 Технологія ADO.NET

Технологія ADO.NET — це технологія роботи з даними, яка базується на платформі .NET Framework. Об'єктна модель ADO.NET надає дві множини класів, які виконують роботу з базою даних:

- класи під'єднаних об'єктів;
- класи від'єднаних об'єктів.

Класи під'єднаних об'єктів забезпечують встановлення з'єднання з базою даних та управління базою зі сторони додатку.

До них належать Connection, Transaction, DataAdapter, Command, DataReader, Parameter, tableAdapterManager, xxxxTableAdapter.

Клас Connection дозволяє встановлювати з'єднання з джерелом даних. Перед тим, як почати взаємодію з даним необхідно встановити з'єднання.

Клас DataReader дозволяє читати дані, які повертає команда Select, по одному рядку за раз в однонапрямленому потоці, що доступний тільки для читання. Це найпростіший спосіб отримання даних.

Клас Command дозволяє виконати SQL-оператор будь-якого типу.

Адаптер даних DataAdapter визначає чотири властивості: SelectCommand, InsertCommand, UpdateCommand та DeleteCommand. Адаптери даних відправляють зміни назад в базу даних для обробки.

Класи від'єднаних об'єктів забезпечують збереження, використання та перетворення отриманої від бази даних інформації на стороні додатку.

До цих класів належать класи DataSet, DataTable, DataView, DataGridView, DataRow, DataColumn, xxxBindingSource, xxxBindingNavigator.

Тип DataSet — це контейнер для будь-якої кількості об'єктів DataTable, кожен з яких містить колекцію об'єктів DataRow та DataColumn. Як тільки викликаючий процес отримає об'єкт DataSet, викликаючий рівень повністю відключається від бази даних та залишається з локальною копією віддалених даних.

По своїй суті об'єкти DataSet імітують постійне підключення клієнтів, хоча насправді вони працюють з базою даних, що знаходиться в пам'яті.

В під'єднаному режимі на робочому комп'ютері користувача встановлюється термінал або клієнт-серверний додаток, за допомогою якого користувач під'єднується до БД. У від'єднаному режимі для доступу до даних застосовується Web-оглядач [13].

### **3.8 Висновки до розділу 3**

У розділі номер 3 було розглянуто засоби розробки інформаційної системи ефективного менеджменту роботи закладів післядипломної освіти.

До створення програмного продукту найважливіше — провести аналіз існуючих засобів та обрати ті, з якими поставлену задачу можна виконати найшвидше.

У даному випадку, такими засобами є Microsoft SQL Server — для створення бази даних, Microsoft Visual Studio 2017 — як середовище розробки, технологія ADO.NET, бібліотеки System.Data.SqlClient, Microsoft.Office.Interop.Excel та System.Data.SqlClient — для написання коду та Windows Forms — для створення інтерфейсу.

## 4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

В даному розділі йдеться про процес програмування додатку. Наведено UML-діаграми, описуються різні етапи створення програмного продукту: від проектування бази даних в Microsoft SQL Server до інтерфейсу.

### 4.1 Діаграми прецедентів

Діаграми прецедентів (діаграми варіантів) створені перш за все для того, щоб описувати функціональне призначення системи, тобто показувати, що користувач може робити із системою. При цьому механізм функціонування системи від користувача приховано і на діаграмі прецедентів не показується.

Діаграма прецедентів — це вихідне концептуальне представлення або концептуальна модель системи у процесі її проектування та розробки.

Кінцевий користувач діаграми показується як фігурка людини і називається актором. Актором може бути як людина, так і інша технічна система, прилад, програма, яка може впливати на розроблену систему.

Варіанти взаємодії з системою виписуються у окремі еліпси з текстовим варіантом використання та може доповнюватися сценарієм.

Мета варіанта використання — визначити фрагмент поведінки певної сутності без розкриття внутрішньої структури цієї сутності [14].

Також в UML є декілька типів відношення між акторами та прецедентами:

- асоціації (association relationship);
- узагальнення (generalization relationship);
- включення (include relationship);
- розширення (extend relationship).

Найбільш розповсюдженими відношеннями для діаграм прецедентів, що будуть наведені нижче, є включення, розширення та асоціації.



На рисунку 4.1 представлено діаграму прецедентів для одного з вікон Windows Forms інформаційної система, а саме “Оформлення розкладу”.

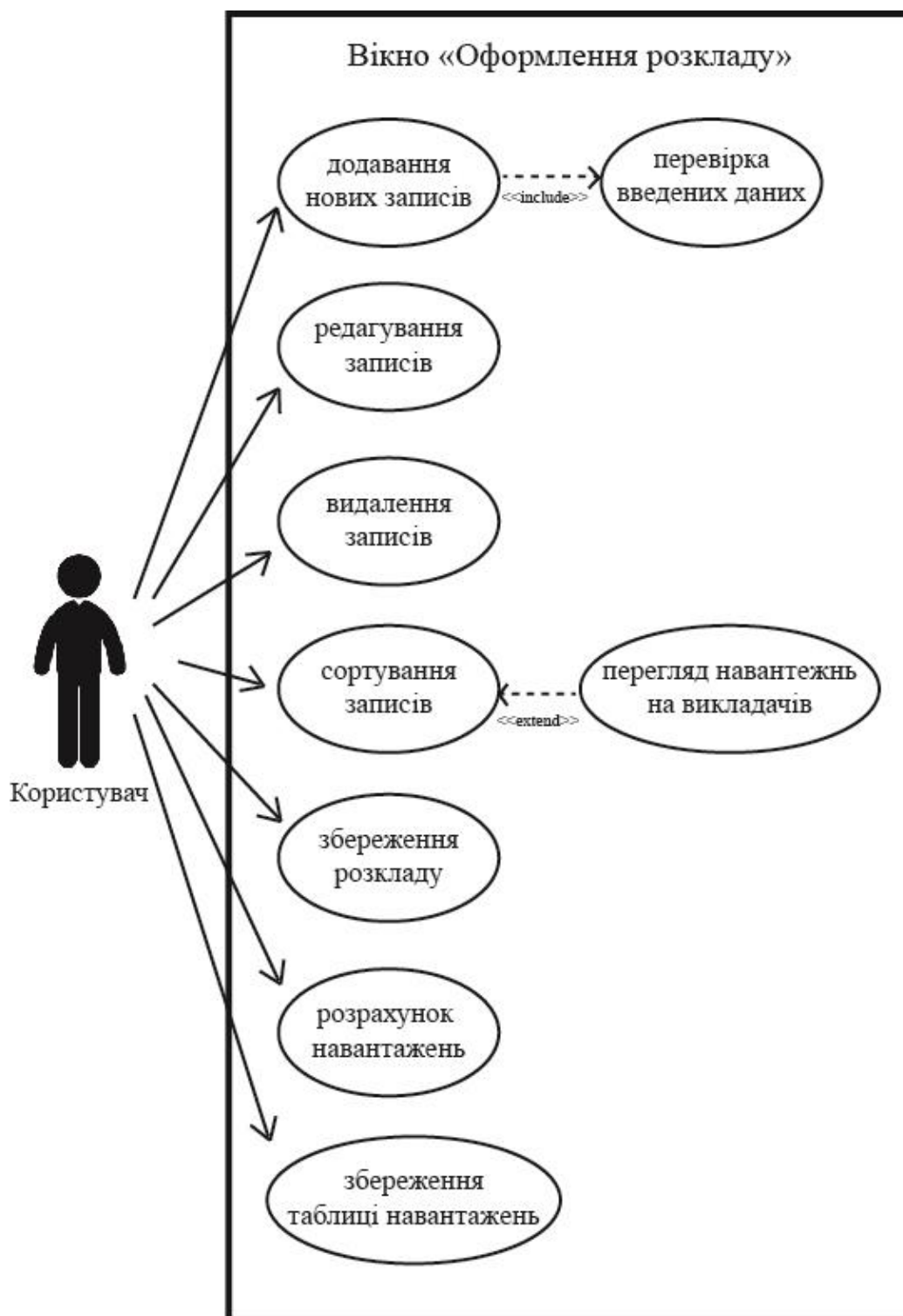


Рисунок 4.1 — діаграма прецедентів для вікна “Оформлення розкладу”

На діаграмі можна побачити найважливіші варіанти користування з вікном.

Відношенням включення пов'язані варіанти “додавання нових записів” та “перевірка введених даних”, тому що користувач має перевірити дані, перш ніж занести їх до таблиці.

Відношенням розширення пов'язані варіанти “сортування записів” та “перегляд навантажень на викладачів”, тому що останній варіант є необов'язковим, користувач звертається до нього за бажанням.

На рисунку 4.2 показано діаграму прецедентів для вікна програми “Редагування викладачів”. У вікон “Редагування аудиторій”, “Редагування спеціалізацій” та “Редагування груп” діаграми прецедентів аналогічні.

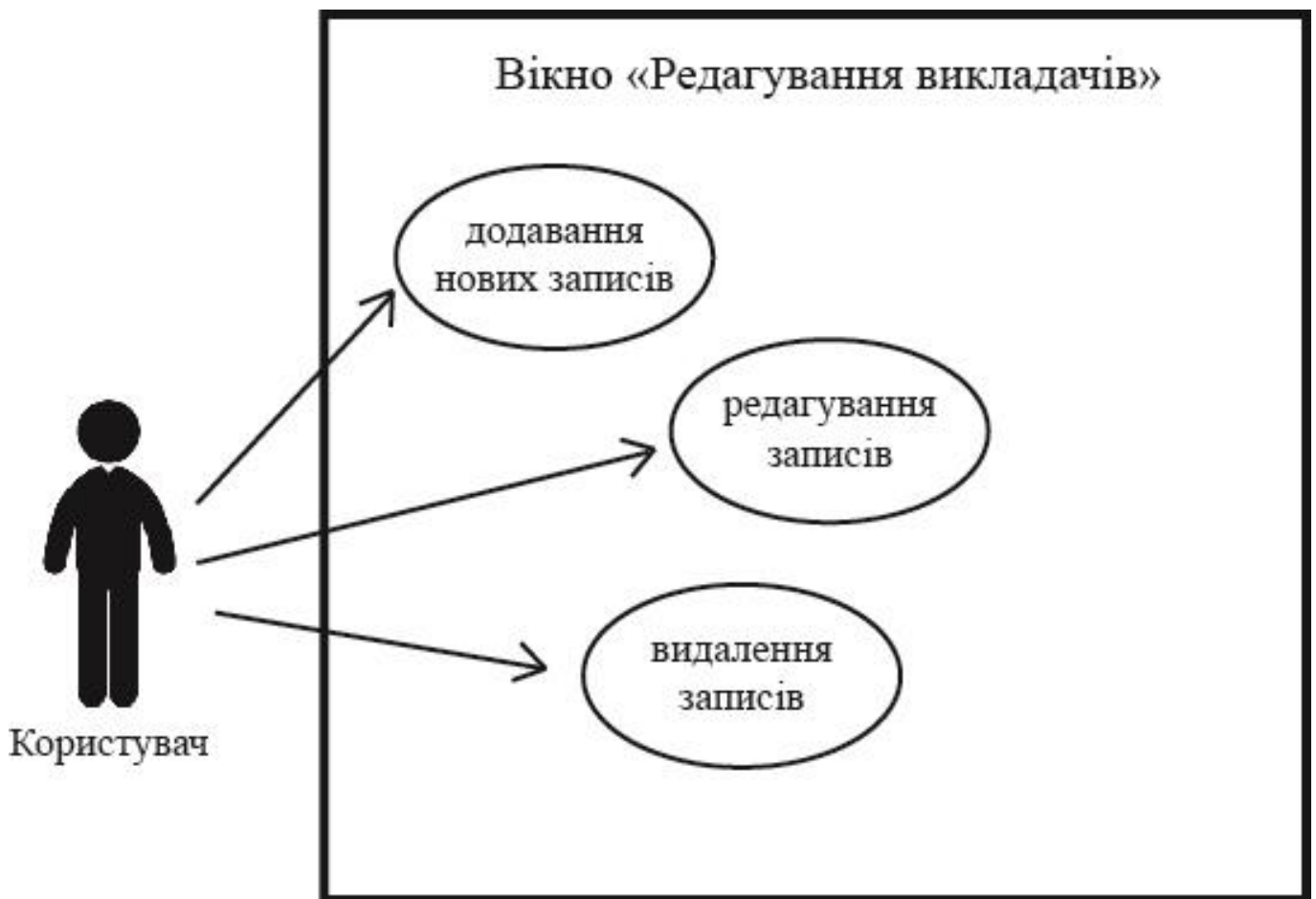


Рисунок 4.2 — діаграма прецедентів для вікна “Редагування викладачів”

Вікно “Редагування предметів” відрізняється від інших вікон редагування даних тим, що там наявний функціонал сортування записів та можна перейти до іншого вікна — “Редагування спеціалізацій”.

На рисунку 4.3 наведено діаграму прецедентів вікна “Редагування предметів”:

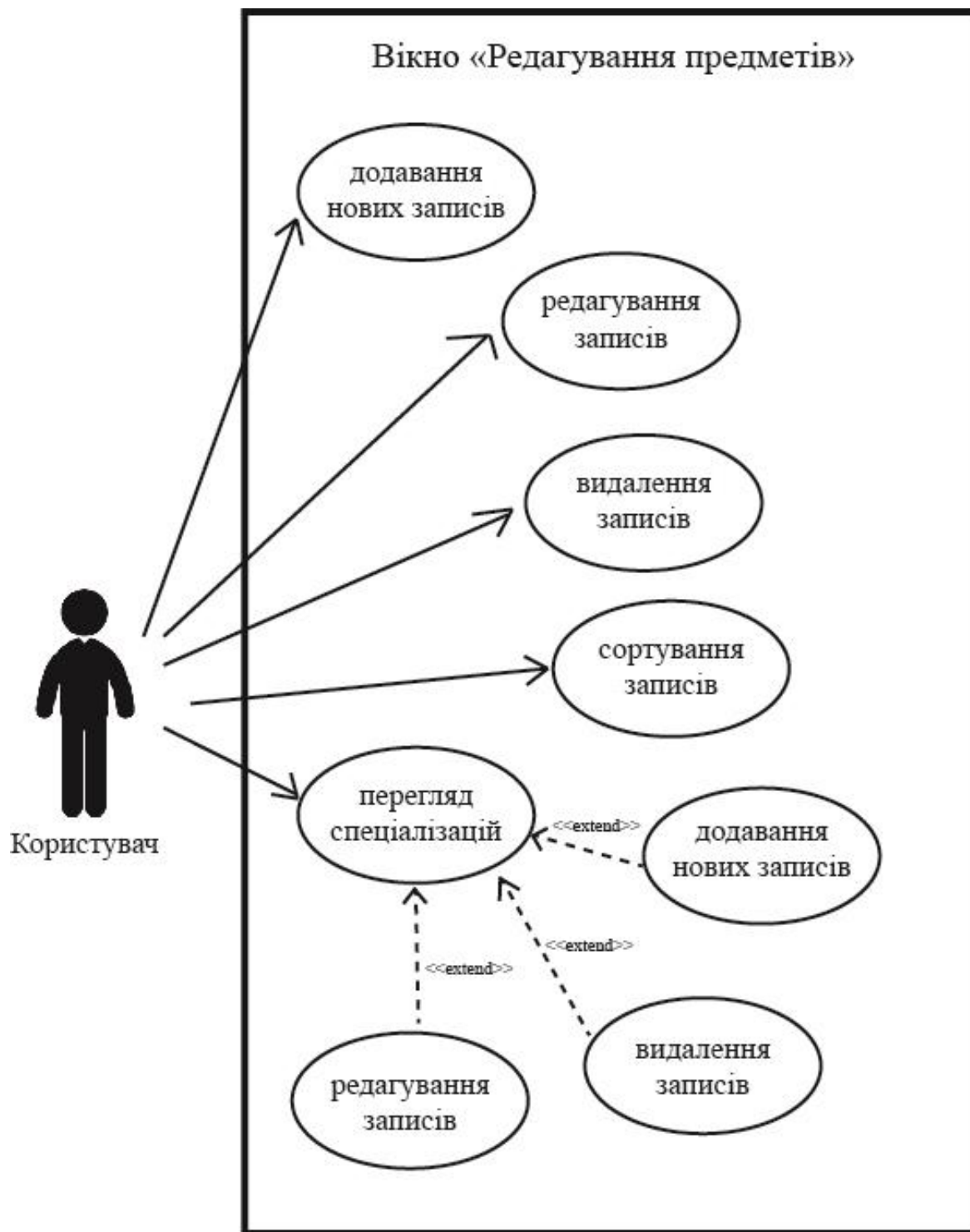


Рисунок 4.3 — діаграма прецедентів для вікна “Редагування предметів”

На цьому розгляд основних вікон модуля формування розкладу інформаційної системи ефективного менеджменту роботи закладів післядипломної освіти завершено.

На рисунку 4.4 наведено діаграму прецедентів модуля менеджменту задач.

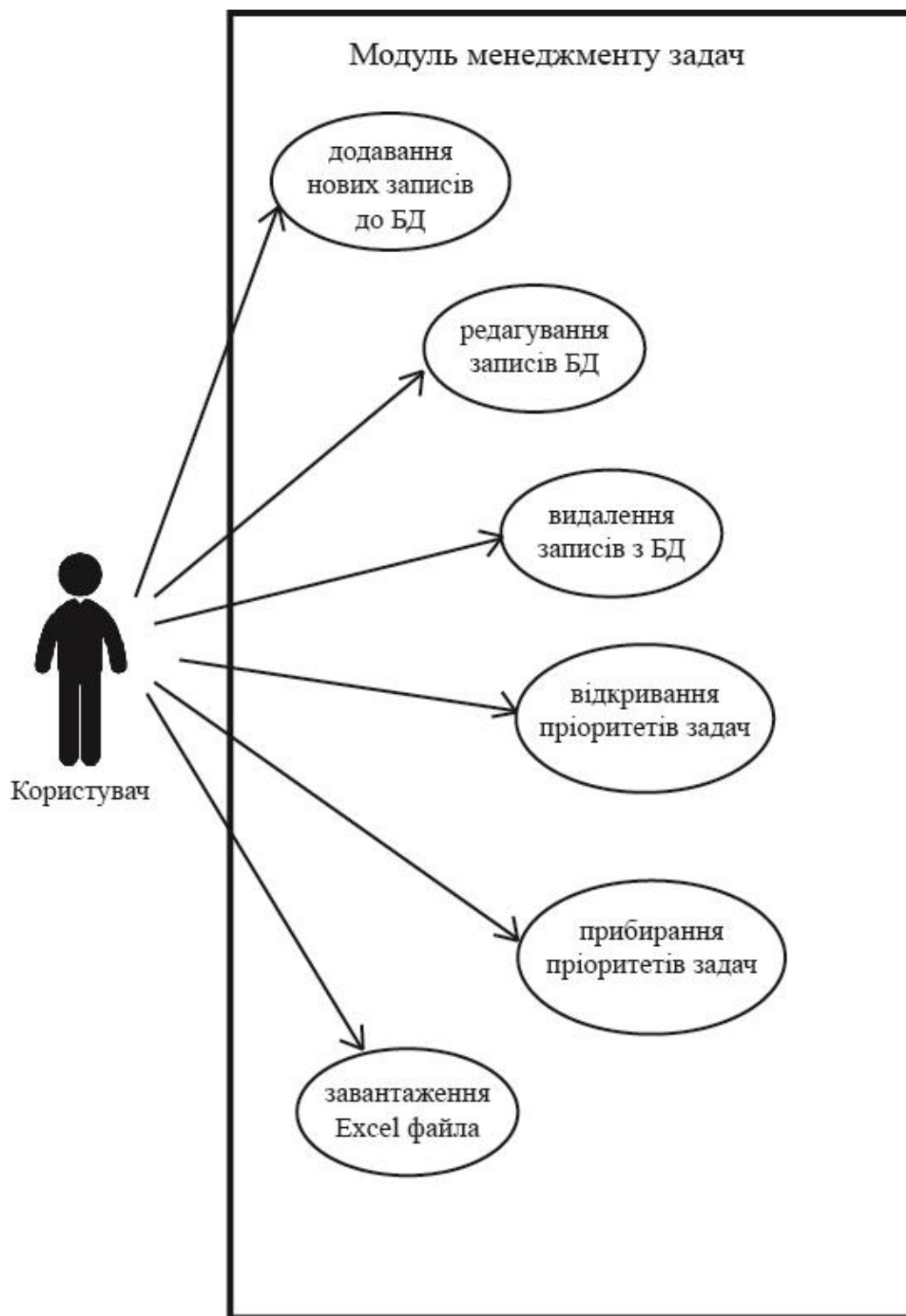


Рисунок 4.4 — діаграма прецедентів модуля менеджменту задач

Діаграми прецедентів є важливим етапом проектування системи, тому що дозволяють чітко бачити, який саме функціонал від неї вимагається.

## 4.2 Створення бази даних в Microsoft SQL Server

### 4.2.1 Створення бази даних для модуля формування розкладу

Всього у Microsoft SQL Server було створено 8 таблиць:

- Teacher;
- Room;
- Lesson;
- Specialization;
- TypeSubj;
- Subject;
- Groups;
- scheduleVarTest.

В усіх таблицях наявне поле `id`, якому надано властивості первинного ключа. Первинний ключ — це один з прикладів унікального індексу і використовується для унікальної ідентифікації записів таблиці.

У даному випадку первинні ключі надані перш за все для ідентифікації записів та полегшення подальшої роботи з ними, зокрема для видалення помилкових або непотрібних записів.

Сутність видалення — у визначенні рядків, які мають бути видалені. Дія `DELETE` завжди визначається її предикатом `WHERE`. Якщо забрати `WHERE`, то видалиться уся таблиця.

Задаючи предикат `WHERE`, ми можемо зменшити область дії `DELETE` до певної групи рядків або до одного рядка. При видаленні одного рядка він зазвичай ідентифікується за допомогою первинного ключа[15].

Для таблиць `Specialization` та `TypeSubj` наявність первинного ключа також є важливою, тому що він використовується для зв'язку цих таблиць з таблицею `Subject`. В цьому випадку застосовано зв'язок “один до багатьох”. Таблиці `Specialization` та `TypeSubj` є батьківськими, а таблиця `Subject` — дочірньою, тому що поля в ній залежать від полів інших двох таблиць.

Сама структура бази даних виглядає наступним чином (рисунок 4.5):

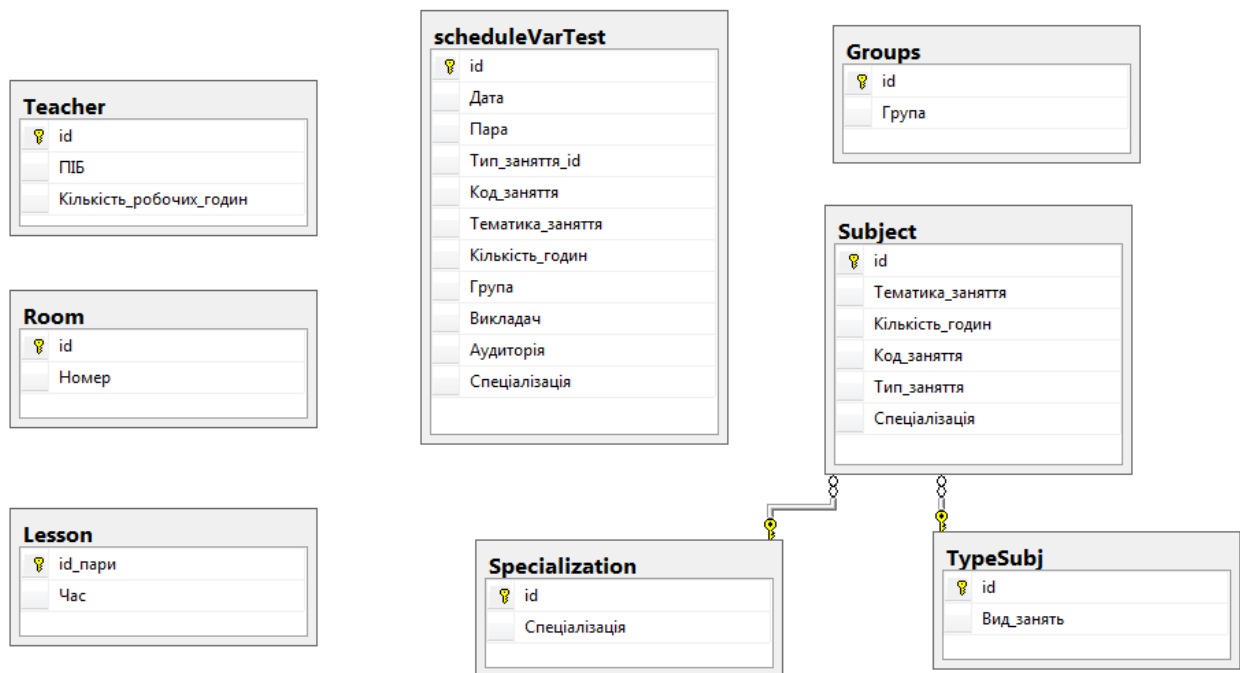


Рисунок 4.5 — Структура бази даних

В таблиці Subject всього 6 полів:

- id;
- Тематика\_заняття;
- Ккількість\_годин;
- Код\_заняття;
- Тип\_заняття;
- Спеціалізація.

Поле Код\_заняття відповідає за шифр теми заняття, наприклад СЗ означає, що це третє семінарське заняття.

Поле Тип\_заняття — робоче поле, яке було введено для того, щоб в подальшому можна було визначити, чи накладаються обмеження “дві групи не можуть бути в одній аудиторії” та “один викладач не може викладати одночасно в двох групах”.

Поле Спеціалізація було вирішено додати для того, щоб була можливість в подальшому відсортувати предмети. У НМАПО ім. П.Л. Шупіка предмети читаються спеціалізацій, проте невідомо наперед, чи буде створено нову спеціалізацію. У разі, якщо б було створено таблиці з предметами під кожен спеціалізацію через деякий час користувач може стикнутися з проблемою, що утворили нову спеціалізацію, але таблиці для неї не передбачено. Наразі усі предмети зберігаються в одній таблиці, але для кожного предмета додано поле, в якому вказано, до якої спеціалізації він відноситься. Структуру таблиці Subject наведено нижче (рисунок 4.6).


	Column Name	Data Type	Allow Nulls
	id	int	<input type="checkbox"/>
	Тематика_заняття	varchar(5000)	<input checked="" type="checkbox"/>
	Кількість_годин	int	<input checked="" type="checkbox"/>
	Код_заняття	varchar(50)	<input checked="" type="checkbox"/>
	Тип_заняття	int	<input checked="" type="checkbox"/>
	Спеціалізація	int	<input checked="" type="checkbox"/>

Рисунок 4.6 — Структура таблиці Subject

Таблиця scheduleVarTest по своїй сутності є головною таблицею, в якій відображається розклад. Вона містить наступні поля:

- id;
- Дата;
- Пара;
- Тип\_заняття\_id;
- Код\_заняття;
- Тематика\_заняття;
- Кількість\_годин;
- Група;
- Викладач;
- Аудиторія;

— Спеціалізація.

До таблиці не було проведено зовнішніх зв'язків за допомогою засобів Microsoft SQL Server, тому що пізніше було зроблено необхідну симуляцію зв'язків за допомогою засобів Windows Forms. Структуру таблиці scheduleVarTest наведено нижче (рисунок 4.7).


	Column Name	Data Type	Allow Nulls
	id	int	<input type="checkbox"/>
	Дата	datetime	<input checked="" type="checkbox"/>
	Пара	int	<input checked="" type="checkbox"/>
	Тип_заняття_id	int	<input checked="" type="checkbox"/>
	Код_заняття	varchar(50)	<input checked="" type="checkbox"/>
	Тематика_заняття	varchar(5000)	<input checked="" type="checkbox"/>
	Кількість_годин	int	<input checked="" type="checkbox"/>
	Група	varchar(50)	<input checked="" type="checkbox"/>
	Викладач	varchar(200)	<input checked="" type="checkbox"/>
	Аудиторія	varchar(50)	<input checked="" type="checkbox"/>
	Спеціалізація	varchar(200)	<input checked="" type="checkbox"/>

Рисунок 4.7 — Структура таблиці scheduleVarTest

Поле id призначене для унікальної ідентифікації запису. В поле Дата типу datetime заноситься дата заняття, а в поле Пара — порядковий номер пари. У поля Тип\_заняття\_id, Код\_заняття, Тематика\_заняття та Кількість\_годин будуть записуватися відповідні дані, які на етапі проектування форми Windows Forms будуть заноситися з таблиці Subject.

У поле Група типу varchar(50) заноситься код групи, у поле Викладач — прізвище та ініціали викладача, у поле Аудиторія — номер або шифр аудиторії та у поле Спеціалізація — назва спеціалізації.

Усі ці дані також будуть підтягуватися з відповідних таблиць на моменті роботи з Windows Forms.



Структуру таблиць Teacher, Room, Groups, Lesson наведено на рисунку 4.8. У цих таблицях первинний ключ було додано задля ідентифікації записів та коректного їх видалення, а для зовнішніх зв'язків вони не застосовуються.

Teacher		
	Column Name	Data Type
🔑	id	int
	ПІБ	varchar(200)
	Кількість_робочих_годин	int

Room		
	Column Name	Data Type
🔑	id	int
	Номер	varchar(50)

Groups		
	Column Name	Data Type
🔑	id	int
	Група	varchar(50)

Lesson		
	Column Name	Data Type
🔑	id_пари	int
	Час	varchar(50)

Рисунок 4.8 — Структура таблиць

#### 4.2.2 Створення бази даних для модуля менеджера задач

Для цього модуля було створено одну таблицю — plan (рисунок 4.9).

	Column Name	Data Type	Allow Nulls
🔑	Номер_п_п	int	<input type="checkbox"/>
	Рішення_вченої_ради	varchar(500)	<input type="checkbox"/>
	Дата_та_номер_проток...	varchar(200)	<input type="checkbox"/>
	Термін_виконання	datetime	<input type="checkbox"/>
	Відповідальні_особи_за...	varchar(500)	<input type="checkbox"/>
	Етап_виконання	varchar(1000)	<input type="checkbox"/>
	Виконано	varchar(200)	<input type="checkbox"/>

Рисунок 4.9 — структура таблиці plan

Полю Номер\_п\_п було надано первинний ключ для ідентифікації записів.

Полю Термін\_виконання надано тип datetime, тому що саме від цього поля залежить робота модуля менеджера задач.

### 4.3 Підключення бази даних модуля формування розкладу до проекту Visual Studio та робота з таблицями

Завдяки засобам роботи Visual Studio 2017 з базами даних, створених за допомогою SQL Server базу даних було додано до джерела даних проекту.

На формі “Оформлення розкладу” було створено DataGridView, який відображає дані, які вже збережені до бази даних, але не дозволяє додавати нові записи. Додавати записи до DataGridView можна за допомогою елементів управління comboBox та textBox. Дані, які відображаються в comboBox витягуються з відповідних таблиць бази даних (рисунок 4.10). Таким чином за допомогою засобів Windows Forms було зроблено симуляцію зовнішніх зв’язків. Користувач не зможе ввести в comboBox, а отже і до DataGridView значення, якого не існує в таблиці бази даних.

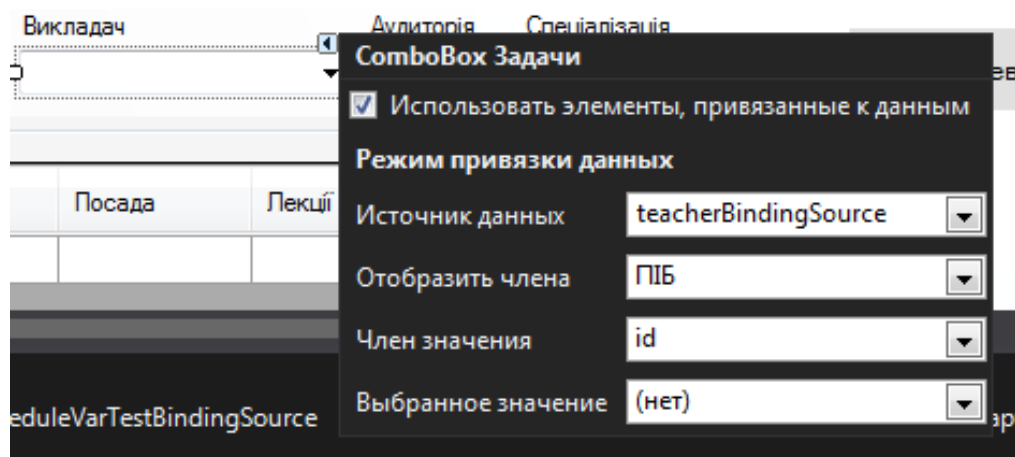


Рисунок 4.10 — Приклад налаштування comboBox Викладач

Додавання даних з елементів управління comboBox та textBox до DataGridView було здійснено завдяки підключенню Odbc та SQL-запитам.

Також на формі знаходяться елементи управління, завдяки яким можна вирахувати навантаження на викладачів в рамках циклу навчання для певної спеціалізації. В коді елементів управління використовуються SQL-запити та підключення Odbc та Sql. Завдяки цим підключенням запити, описані в коді, параметризовані, в якості параметрів вони беруть дані з елементів управління.

Також на формі є кнопки, які відповідають за збереження розкладу та таблиці навантаження викладачів до файлів Excel. В коді це реалізовано за допомогою можливостей бібліотеки Microsoft.Office.Interop.Excel.

Завдяки класам Application, Workbook та Worksheet цієї бібліотеки створюється новий документ Microsoft Excel відповідно з книгою та листами. Далі цикл проходить по усім рядкам в таблиці DataGridView та за допомогою властивості Cells отримує значення комірок, які передає коміркам в документі Excel [16].

При формуванні Excel-файла було опущено робочі поля таблиці DataGridView, такі як id, Тип\_заняття та Спеціалізація, а отже сформована таблиця мала вигляд, який необхідний за нормами.

Завдяки властивостям ColumnWidth, HorizontalAlignment, VerticalAlignment та Style.WrapText задається стиль комірок документа Excel.

По завершенню формування створений документ викликається та відкривається у програмі Microsoft Excel, звідки його можна переглянути та зберегти для подальшого використання.

Перед тим, як додати новий запис до DataGridView з розкладом, користувач має перевірити коректність введених даних, натиснувши на кнопку “Перевірити дані”. В коді цього елемента управління прописано алгоритм, який перевіряє, чи не потрапляють введені дані до обмежень:

- дві групи не можуть бути в одній аудиторії;
- два викладача не може викладати в одній аудиторії;
- одна група не може бути одночасно в двох аудиторіях;
- один викладач не може викладати одночасно в двох аудиторіях;
- один викладач не може викладати одночасно в двох групах;
- два викладача не може викладати одночасно в одній групі.

Але, якщо тип заняття – “Лекція”, тобто полю Тип\_заняття надано значення 1, тоді декілька груп може бути в одній аудиторії та один викладач може одночасно читати більш, ніж в двох групах.

Алгоритм працює наступним чином (блок-схему наведено на рисунку 4.11 та рисунку 4.12):

1. Починається перша ітерація циклу по усім рядкам DataGridView.

2. Дані, введені в textBox з датою, textBox з парою, comboBox з аудиторією, textBox з типом заняття та comboBox з викладачем, перевіряються з відповідними даними з рядка DataGridView. Якщо дані про дату, пару та аудиторію співпадають, дані про викладача різні, а тип заняття — 1, тоді користувач отримує повідомлення, що виникла помилка, на даній лекції вже є викладач.

3. Дані, введені в textBox з датою, textBox з парою, comboBox з викладачем та textBox з типом заняття, перевіряються з відповідними даними з рядка DataGridView. Якщо дані про дату, пару та викладача співпадають, а тип заняття не дорівнює 1, тоді користувач отримує повідомлення, що виникла помилка, у даного викладача вже є пара в цей час.

4. Дані, введені в textBox з датою, textBox з парою, comboBox з аудиторією та textBox з типом заняття, перевіряються з відповідними даними з рядка DataGridView.

Якщо дані про дату, пару та аудиторію співпадають, а тип заняття не дорівнює 1, тоді користувач отримує повідомлення, що виникла помилка, дана аудиторія вже зайнята в цей час.

5. Дані, введені в textBox з датою, textBox з парою, comboBox з групою та textBox з типом заняття, перевіряються з відповідними даними з рядка DataGridView.

Якщо дані про дату, пару та групу співпадають, а тип заняття не дорівнює 1, тоді користувач отримує повідомлення, що виникла помилка, дана група вже має пару в цей час.

6. Дані, введені в textBox з датою, textBox з парою, comboBox з групою, comboBox з викладачем, comboBox з аудиторією та textBox з типом заняття, перевіряються з відповідними даними з рядка DataGridView.

Якщо дані про дату, пару, викладача, аудиторію та групу співпадають, а тип заняття дорівнює 1, тоді користувач отримує повідомлення, що виникла помилка, дана лекція для даної групи вже існує.

7. У всіх інших випадках помилки немає і програма переходить до перевірки даних з наступним рядком DataGridView.

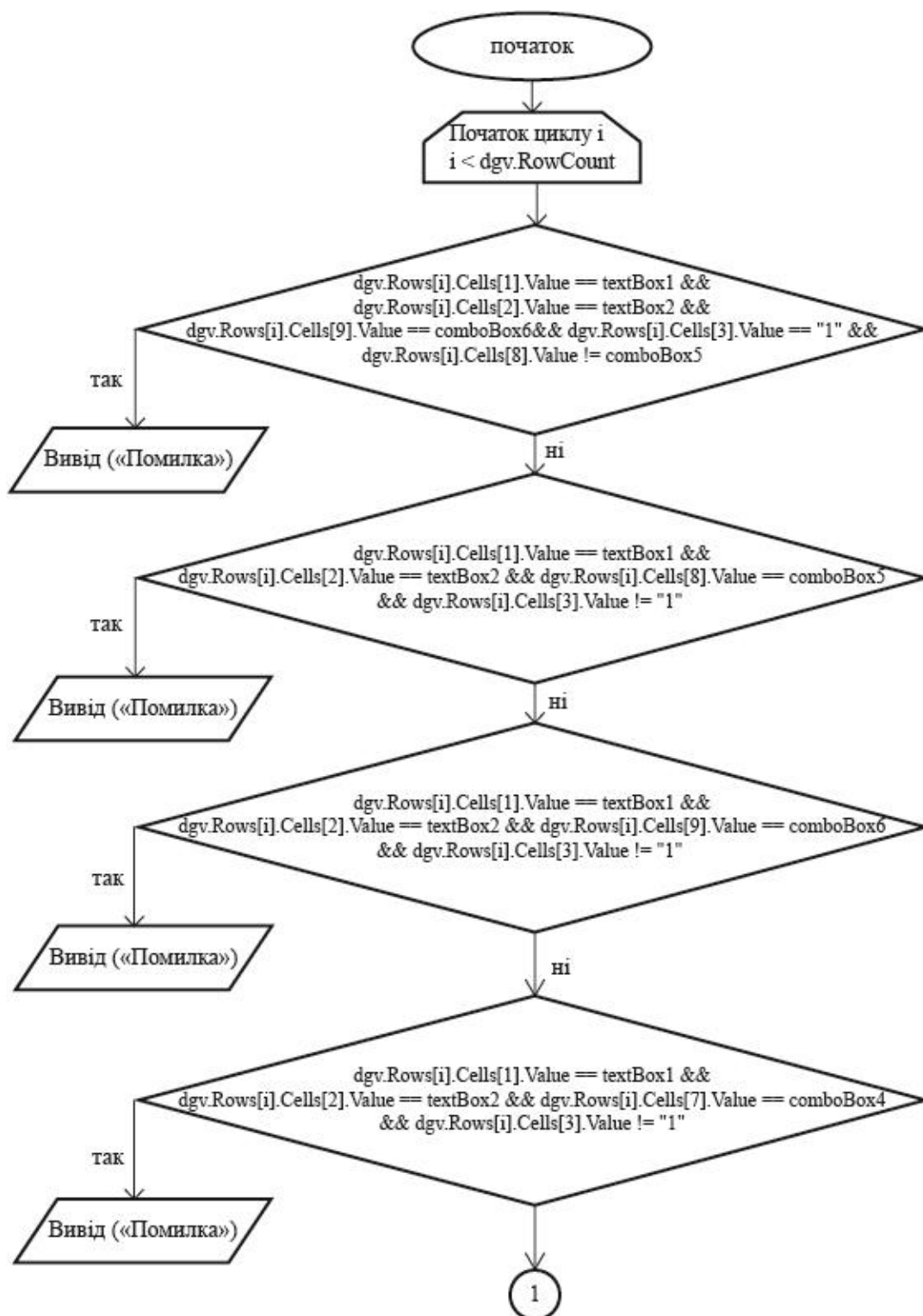


Рисунок 4.11 — початок блок-схеми алгоритму

На рисунку 4.11 зображено першу частину блок-схеми алгоритму.

Продовження блок-схеми:

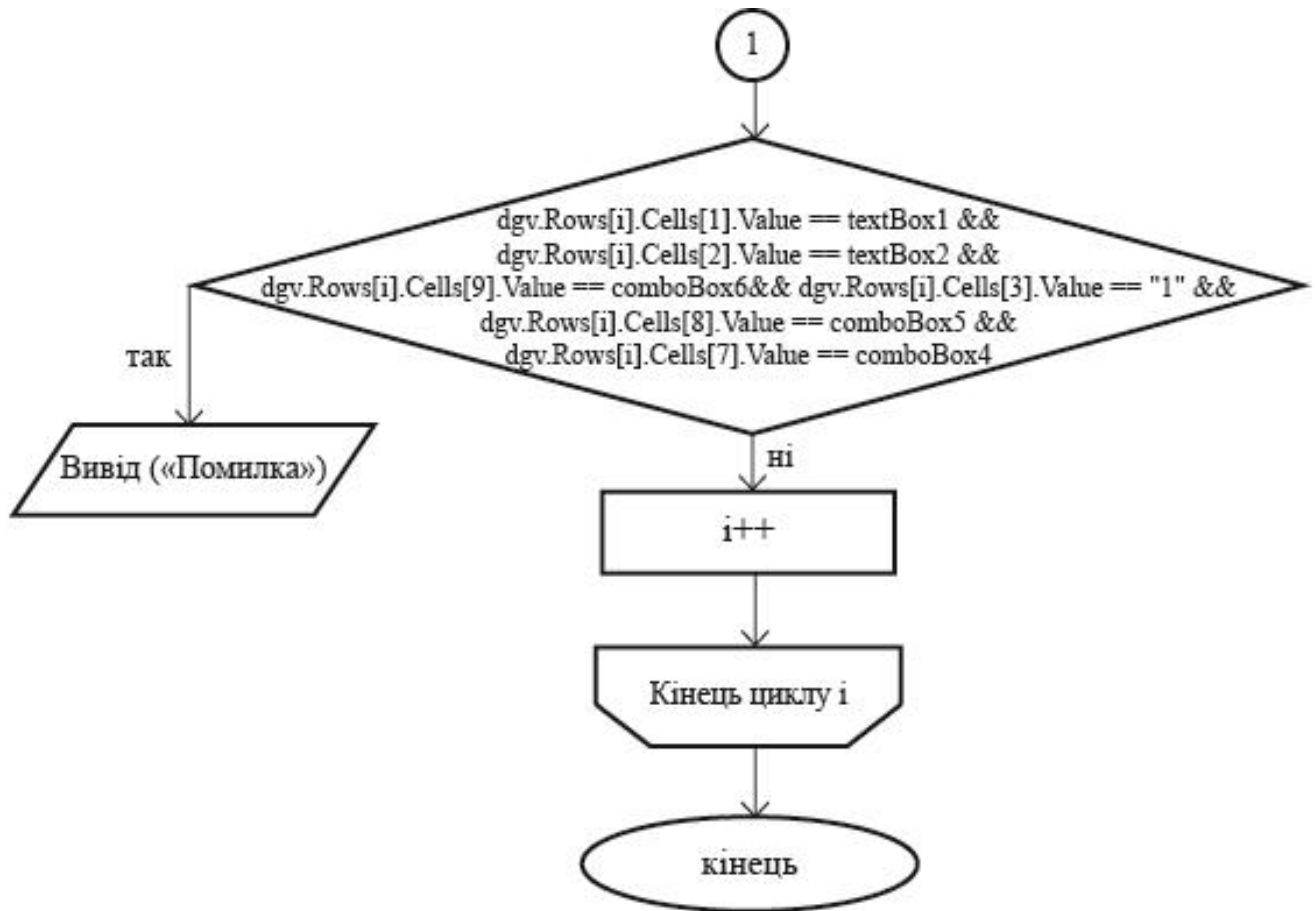


Рисунок 4.12 — продовження блок-схеми алгоритму

На різних формах проекту “Редагування предметів”, “Редагування викладачів”, “Редагування груп” та “Редагування аудиторій” було створено відповідні DataGridView, які відображають зміст таблиць на форму та дозволяють додавати нові записи та редагувати та видаляти наявні.

#### 4.4 Підключення бази даних модуля менеджера задач до проекту Visual Studio та робота з таблицями

Аналогічними засобами ADO.NET, розглянутими у пункті вище, до джерела даних було додано базу даних, що містить таблицю plan.

Основна задача розробленого менеджера задач — якимось чином показувати, скільки часу залишилось до виконання задач. Було обрано наступний варіант:

— якщо до строку задачі залишається більше 30 днів — задача виділяється зеленим кольором;

— якщо до строку задачі залишається більше 15, але менше 30 днів — вона виділяється жовтим кольором;

— якщо залишилось менше 15 днів — задача стає червоною.

Це було реалізовано за допомогою циклу, який проходить по усім рядкам `DataGridView` з планом робіт. В циклі створюється змінна, яка відповідає за кількість днів до дедлайну. Вона вираховується наступним чином:

— береться значення з колонки `DataGridView`, яка відповідає за дату виконання задачі та приводиться до типу `string`;

— за допомогою функції `DateTime.Parse()` строкова змінна перетворюється на структуру `DateTime`, яка по факту є датою;

— від цієї дати віднімається поточна дата. Це реалізовано за допомогою властивості `DateTime.Now`, яка бере поточну дату та час комп'ютера [17];

— за допомогою властивості `Days` вираз віднімання дат перетворюється на тип `Int32`, який в подальшому порівнюється з правилами виділення задач кольором, що описані вище.

## 4.5 Висновки до розділу 4

У розділі номер 4 було розглянуто етапи розробки системи ефективного менеджменту закладів післядипломної освіти.

Було описано яким чином створено базу даних, які структури мають таблиці та чому вони саме такі. Розглянуто, як створено проект з цими джерелами даних та які елементи управління додано. Наведено алгоритм перевірки даних, які заносяться до блоку формування розкладу. Описано деякі з методів, які використовувалися під час написання коду.

## **5 РОБОТА КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ**

В цьому розділі наведено системні вимоги для роботи з додатком та сценарії роботи користувача з ним.

### **5.1 Системні вимоги**

Для забезпечення безперебійної роботи інформаційної системи ефективного менеджменту роботи закладів післядипломної освіти комп'ютер, на який встановлюється програма, повинен відповідати наступним вимогам:

1. Операційна система Windows 10 редакції “Домашня”, Pro або “Корпоративна”/ Windows 7 редакції “Домашня”, “Професійна”, “Корпоративна” або “Максимальна” з пакетом оновлень Service Pack 1.
2. Вільна пам'ять на жорсткому диску 150 Мб.
3. Процесор Intel Core i3 і вище з тактовою частотою 1.5 Гц.
4. Відеокарта Intel HD Graphics 4000 і вище.
5. ОЗП 4 Гб.
6. Файлова система NTFS. Програму також можна встановити на комп'ютер з файловою системою FAT32, проте вона є менш безпечною.

Також необхідно встановити наступні програми:

1. Microsoft SQL Server 2014 Express — для створення локального сервера, на якому буде зберігатися база даних проекту.
2. Microsoft Excel — тому що програмний додаток підтримує зчитування вхідних даних з документів Excel та зберігає вихідні дані у Excel-файли.
3. .NET Framework — для коректної роботи Microsoft SQL Server. Фреймворк можна автоматично встановити під час завантаження Microsoft SQL Server.



4. Microsoft Visual Studio Community 2017 — у випадку, якщо в подальшому буде необхідно внести правки в проект. Також потрібно, щоб були встановлені наступні компоненти: Розробка класичних додатків .Net, Розробка додатків для універсальної платформи Windows, Зберігання та обробка даних та SQL Search.

## 5.2 Робота з програмним модулем для складання розкладу

Початкове вікно Windows Forms для роботи з додатком “Інформаційна система для ефективного менеджменту закладу післядипломної освіти” (рисунок 5.1) має п’ять кнопок: “Редагувати предмет”, “Редагувати викладача”, “Редагувати аудиторії”, “Редагувати групи” та “Розклад”, при натисканні на які користувач буде переходити до відповідних вікон.



Рисунок 5.1 — Головне вікно додатку Windows Forms

Користувач може обрати, до якого вікна він хоче звернутися, виходячи з того, які саме дані він хоче побачити або відредагувати.

Першою на головному вікні додатку іде кнопка, яка покаже користувачу вікно редагування предметів (рисунок 5.2).

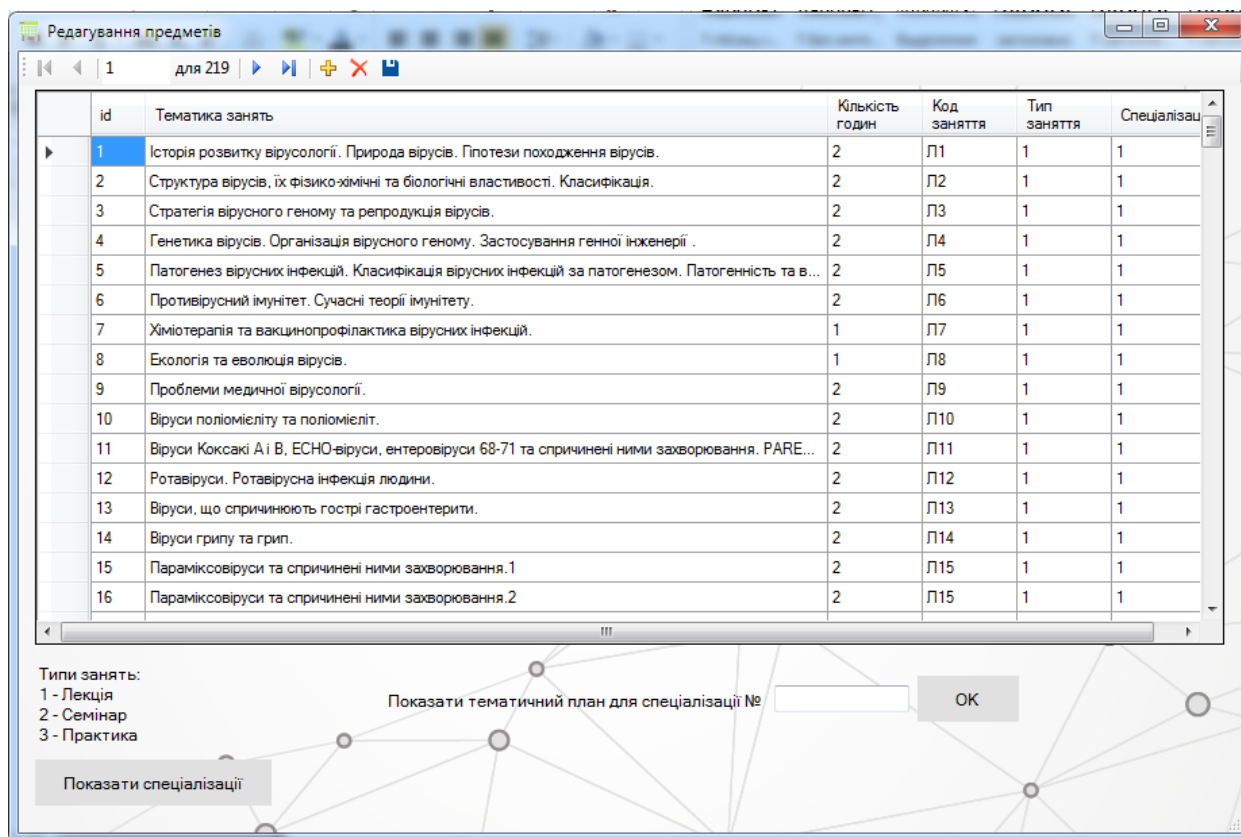


Рисунок 5.2 — вікно редагування предмету

В цьому вікні користувач може додати новий запис, видалити наявний, прокрутитися рядки з записами та зберегти зміни за допомогою елементів управління зверху вікна (рисунок 5.3). В частині, де відображається поточний номер рядка, користувач може ввести конкретний номер рядка і одразу потрапити на нього, не витрачаючи час на скролінг. Також в момент, коли користувач починає вводити новий запис, автоматично генерується ще один рядок для наступного запису, що прирівнюється до кнопки додавання запису зверху вікна. Кожен користувач може користуватися цими елементами, як йому зручніше.

За допомогою `comboBox`, який знаходиться під елементом управління `DataGridView`, користувач може відсортувати предмети для конкретної спеціалізації.

Також під таблицею з предметами знаходиться текстова підказка, яка допомагає користувачу орієнтуватися в тому, якому типу заняття відповідає певний номер.

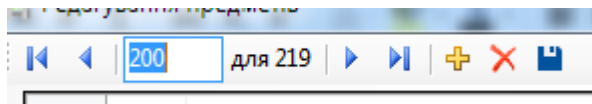


Рисунок 5.3 — елементи управління

Натиснувши на кнопку “Показати спеціалізації” користувач відкриває вікно додатку, де може перевірити id спеціалізації, додати нову або відредагувати наяві (рисунок 5.4). Завдяки властивості вікна TopMost воно відображається зверху вікна з редагуванням предметів.

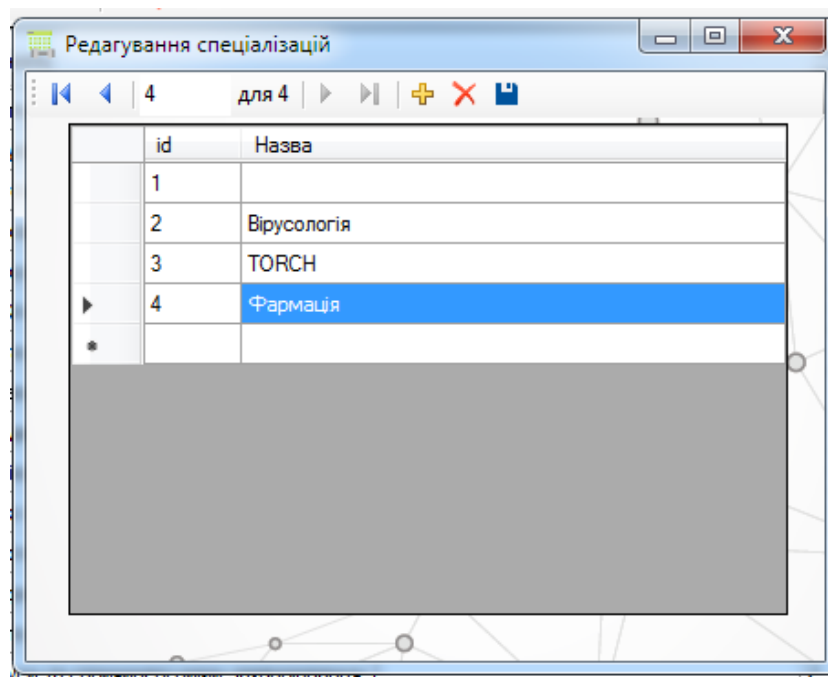


Рисунок 5.4 — вікно редагування спеціалізацій

Завдяки тому, що це вікно відображується зверху інших вікон користувачу буде зручно звіряти id з вікна з редагуванням спеціалізацій та даними зі стовчика Спеціалізація елемента управління DataGridView з вікна редагування предметів (рисунок 5.5).

Також при сортуванні таблиці за спеціалізаціями це допоможе зорієнтуватися, для якої саме спеціалізації відображаються дані.

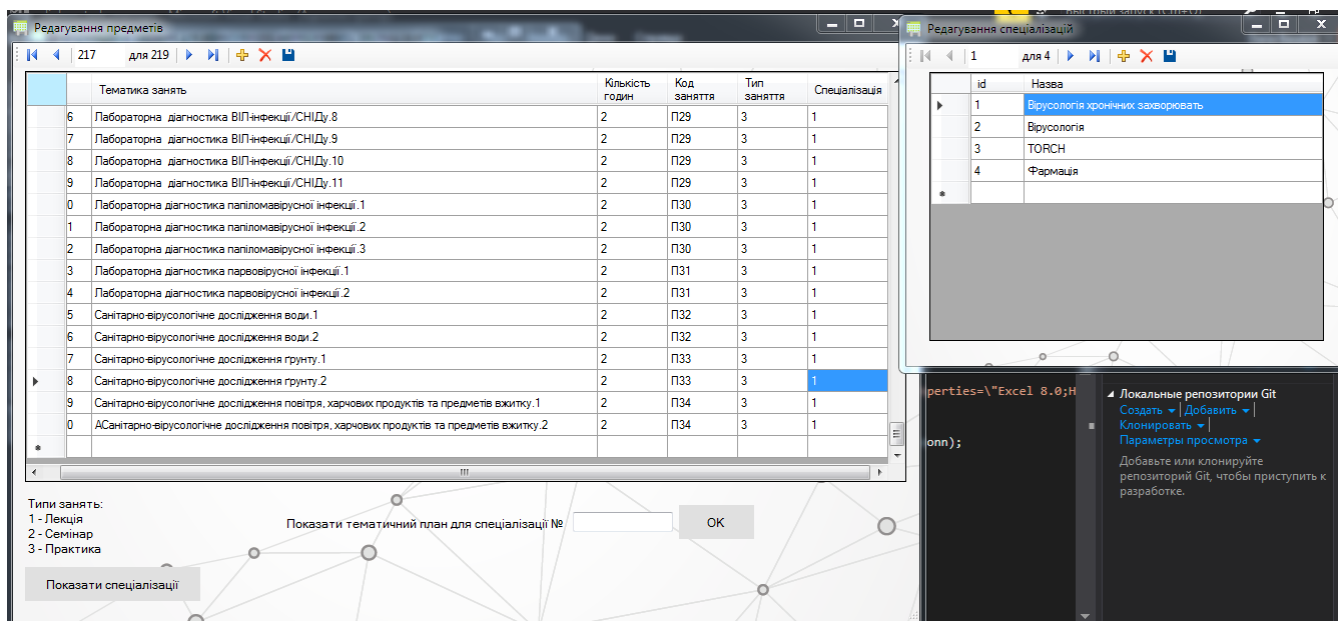


Рисунок 5.5 — одночасна робота у двох вікнах

Інтерфейс вікон “Редагування викладачів”, “Редагування аудиторій” та “Редагування груп” подібний до інтерфейсу вікна “Редагування предметів” і сценарії роботи з ними схожі.

В цих вікнах користувач також може додавати нові записи, видаляти та редагувати наявні. В цих вікнах також є елемент управління, розглянути на рисунку 5.3, що полегшує роботу з записами.

За допомогою усіх цих вікон користувач має заповнити таблиці бази даних актуальними та коректними даними. Це необхідно зробити перш ніж приступати до роботи з вікном “Розклад”, адже безпосередньо в ньому використовуються ці дані.

Вікно додатку “Оформлення розкладу” (рисунок 5.6) має наступні блоки елементів управління:

- елемент управління DataGridView, на якому відображуються записи стосовно розкладу навчання;
- група елементів для сортування даних в таблиці та збереження розкладу до Excel-файлу (рисунок 5.7);

- група елементів для введення даних, які будуть додані до таблиці розкладу та відповідні кнопки для перевірки та додавання даних;
- група елементів управління, яка стосується навантаження на викладачів.

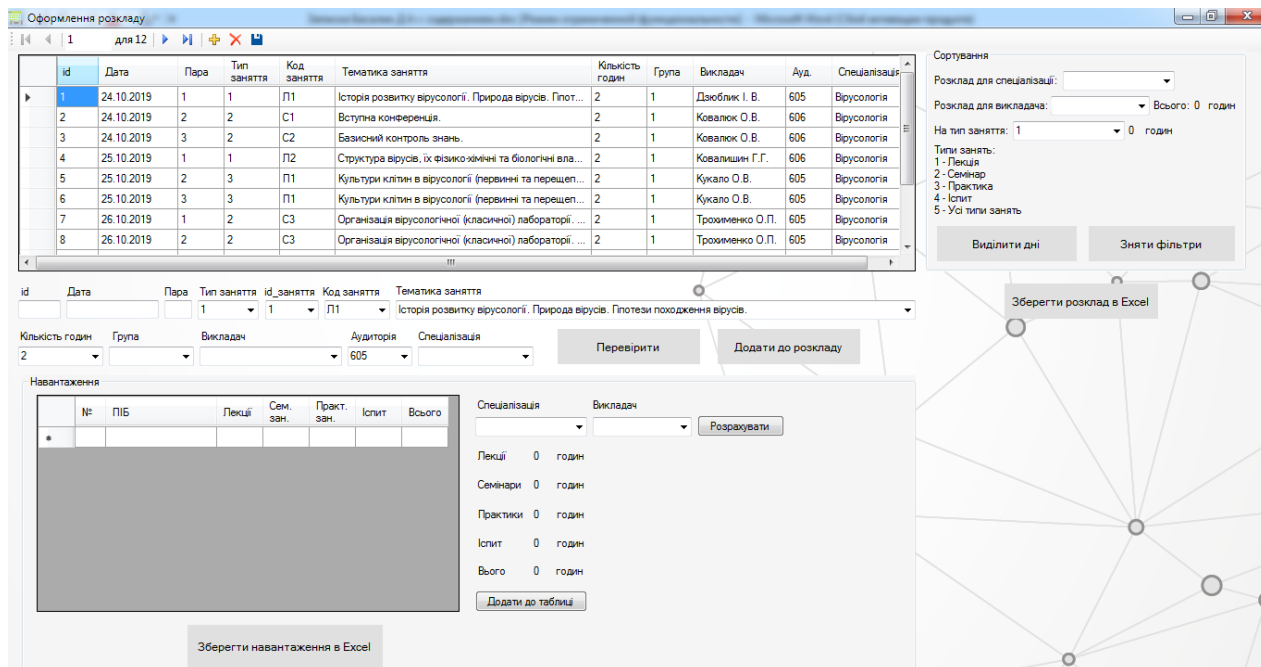


Рисунок 5.6 — вікно “Оформлення розкладу”

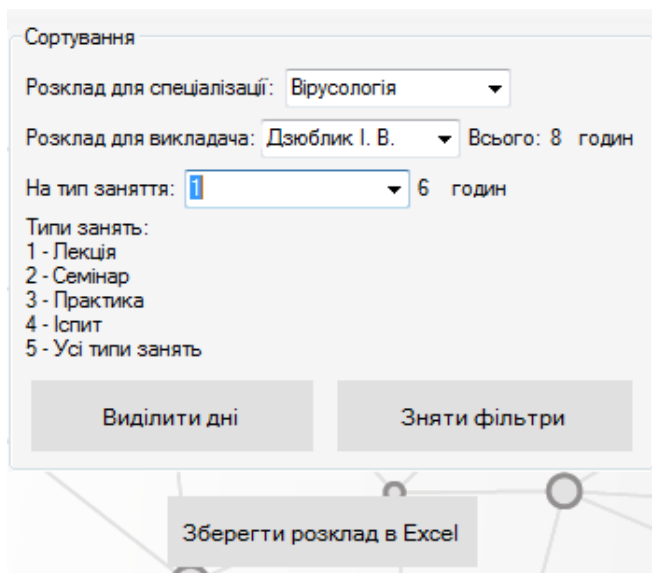


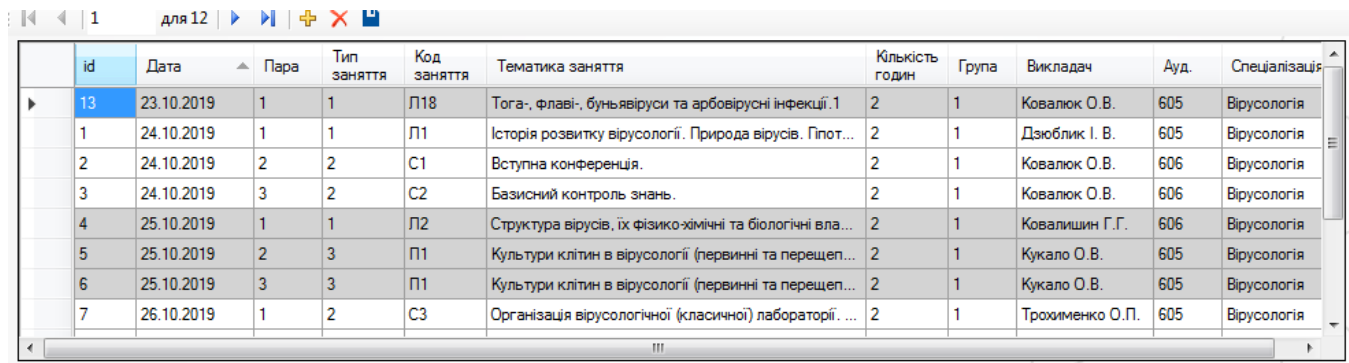
Рисунок 5.7 — вікно приклад взаємодії з блоком сортування

В блоці елементів сортування користувач може обрати для якої саме спеціалізації або якого викладача відобразити розклад. Також користувач може

переглянути загальний розклад для кафедри. Проте важливішим є відображення саме розкладу для певної спеціалізації, тому що саме такий розклад записується до вихідних звітів закладу освіти.

Якщо для заповнення таблиці навантаження користувачеві потрібно дізнатися, скільки годин у викладача вже стоїть на цій спеціальності загалом та скільки годин припадає на кожен тип заняття (лекцію, практику, семінар або іспит), він обирає необхідні дані з comboBox і поруч показуються відповідні дані.

Також користувач може виділити різним кольором парні та непарні дати для більш зручного орієнтування в записах в таблиці (рисунок 5.8).



id	Дата	Пара	Тип заняття	Код заняття	Тематика заняття	Кількість годин	Група	Викладач	Ауд.	Спеціалізація
13	23.10.2019	1	1	П18	Тога-, флаві-, буньявіруси та арбовірусні інфекції.1	2	1	Ковалюк О.В.	605	Вірусологія
1	24.10.2019	1	1	П1	Історія розвитку вірусології. Природа вірусів. Плот...	2	1	Дзюблик І.В.	605	Вірусологія
2	24.10.2019	2	2	С1	Вступна конференція.	2	1	Ковалюк О.В.	606	Вірусологія
3	24.10.2019	3	2	С2	Базисний контроль знань.	2	1	Ковалюк О.В.	606	Вірусологія
4	25.10.2019	1	1	П2	Структура вірусів, їх фізико-хімічні та біологічні вла...	2	1	Ковалишин Г.Г.	606	Вірусологія
5	25.10.2019	2	3	П1	Культури клітин в вірусології (первинні та перещеп...	2	1	Кукало О.В.	605	Вірусологія
6	25.10.2019	3	3	П1	Культури клітин в вірусології (первинні та перещеп...	2	1	Кукало О.В.	605	Вірусологія
7	26.10.2019	1	2	С3	Організація вірусологічної (класичної) лабораторії. ...	2	1	Трохименко О.П.	605	Вірусологія

Рисунок 5.8 — виділення різних днів кольором

За бажанням користувач в будь-який момент може зняти усі фільтри та сортування, натиснувши кнопку “Зняти фільтри”.

Коли користувач заповнить усі дані, які необхідні для формування розкладу, він повинен обрати, для якої спеціалізації сортується розклад та натиснути на кнопку “Зберегти розклад в Excel”.

Після натискання цієї кнопки відкривається програма Microsoft Excel з розкладом, який відсортовано за певною спеціалізацією або розклад для певного викладача (рисунок 5.9). У Microsoft Excel таблиця вже має певне оформлення (вирівнювання даних у комірках зліва або по центру, переноси у комірках по словам, необхідна ширина стовпчиків). Якщо користувач не бажає внести ще якісь зміни вручну, то він може зберігати цей файл як зазвичай.

До розкладу в Excel-файлі записуються лише ті колонки, які вимагаються кафедрою. Робочі колонки id, Тип заняття та Спеціалізація не переносяться.

A	B	C	D	E	F	G	H
Дата	Пара	Код заняття	Тематика заняття	Кількість годин	Група	Викладач	Ауд.
24.10.2019	1	L1	Історія розвитку вірусології. Природа вірусів.	2	1	Дзюблик І. В.	605
24.10.2019	2	C1	Вступна конференція.	2	1	Ковалюк О.В.	606
24.10.2019	3	C2	Базисний контроль знань.	2	1	Ковалюк О.В.	606
25.10.2019	1	L2	Структура вірусів, їх фізико-хімічні та біологічні	2	1	Ковалишин Г.Г.	606
25.10.2019	2	P1	Культири клітин в вірусології (первинні та перещеплювані): приготування, культивування.1	2	1	Кукало О.В.	605

Рисунок 5.9 — фрагмент розкладу, сформованого в Excel-файлі

В групі елементів для введення даних (рисунок 5.10) користувач заповнює відповідні textBox та обирає необхідні дані з comboBox. При виборі необхідного id\_заняття, Коду заняття або Тематики заняття інші дані заповнюються автоматично. Тому рекомендується спочатку заповнити одне з цих полів для економії часу.

Рисунок 5.10 — блок елементів для введення даних

Перш, ніж заносити введені дані до таблиці, користувач має їх перевірити, натиснувши кнопку “Перевірити”.

Для кнопки “Додати до розкладу” написаний код, який перевіряє змінну на значення true і якщо змінна має саме таке значення, тоді спрацьовує код в тілі умовного оператора, який містить функцію, що додає нові дані до DataGridView з розкладом і надає змінній значення false.

Ця змінна приймає значення true тільки після того, як буде натиснуто кнопку “Перевірити”.

Перевіряти дані треба кожного разу, коли користувач хоче додати новий рядок до DataGridView з розкладом.

Якщо він цього не зробить, а одразу натисне на кнопку “Додати до розкладу”, він все одно отримає наступне повідомлення (рисунки 5.11):

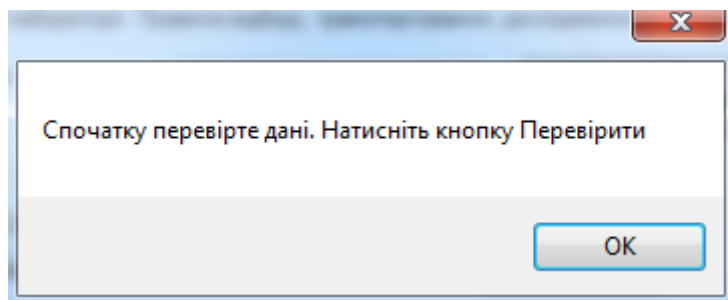


Рисунок 5.11 — повідомлення, якщо дані не були перевірені

Якщо некоректні дані про викладача — наступна помилка (рисунки 5.12):

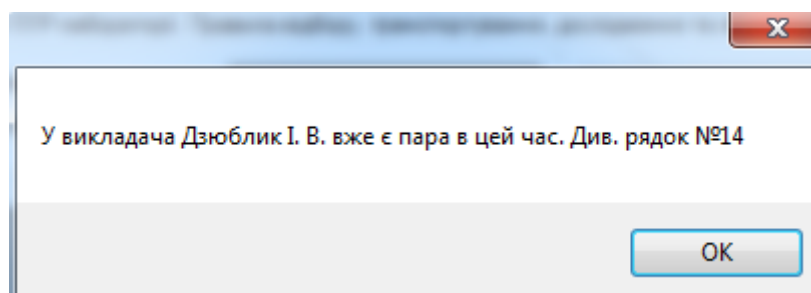


Рисунок 5.12 — повідомлення, що викладач вже зайнятий на цій парі

Якщо некоректна аудиторія, то генерується наступна помилка (рисунки 5.13):

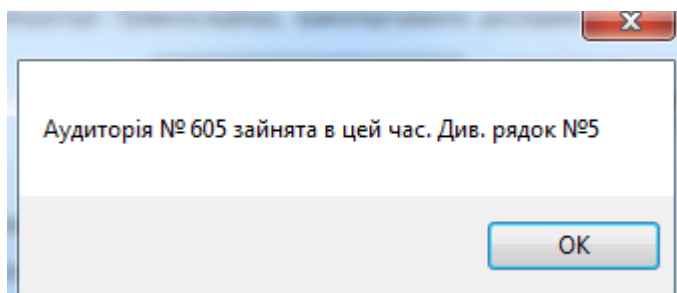


Рисунок 5.13 — повідомлення, що аудиторія вже зайнята

Також можуть виявитися наступні помилки:

— у групи вже є пара (рисунки 5.14);



- на лекцію, де вже є викладач, призначається ще один (рисунок 5.15);
- користувач намагається створити лекцію з такою ж групою, викладачем та аудиторією та на тій же парі, які вже є в розкладі (рисунок 5.16)

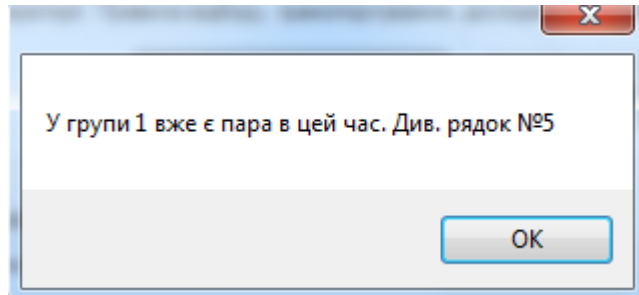


Рисунок 5.14 — повідомлення, що група вже зайнята

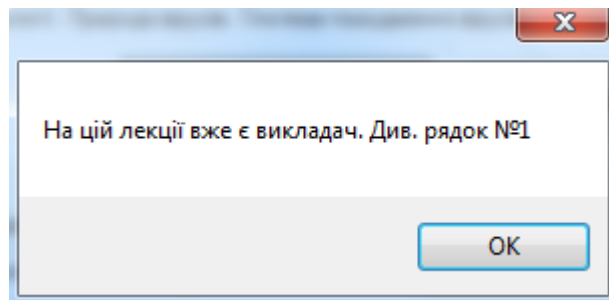


Рисунок 5.15 — повідомлення, що на лекції вже є викладач

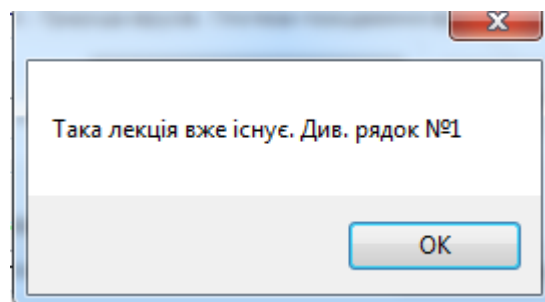


Рисунок 5.16 — повідомлення, що на лекція вже існує

Якщо користувач бачить таку помилку, він має подивитися дані в рядку, з яким знайдено співпадіння і виправити дані в таблиці або в елементах введення даних. Після цього він має перевірити дані ще раз, якщо більше не показуються повідомлення про помилку, це означає, що дані коректні і можна натискати кнопку “Додати до розкладу”.

Блок навантаження на викладачів виглядає наступним чином (рисунок 5.17):

Навантаження

№	ПІБ	Лекції	Сем. зан.	Практи. зан.	Іспит	Всього
*						

Спеціалізація: Вірусологія  
Викладач: Дзюблик І. В.  
Розрахувати

Лекції 0 годин  
Семінари 0 годин  
Практики 0 годин  
Іспит 0 годин  
Всього 0 годин

Додати до таблиці

Зберегти навантаження в Excel

Рисунок 5.17 — блок навантаження на викладачів

В цьому блоці користувач має обрати для якої спеціалізації та для якого викладача вираховується навантаження. Потім він має натиснути кнопку “Розрахувати” і буде проведено розрахунок даних, які беруться з DataGridView з розкладом та записуються у відповідні поля блоку навантаження (рисунок 5.18).

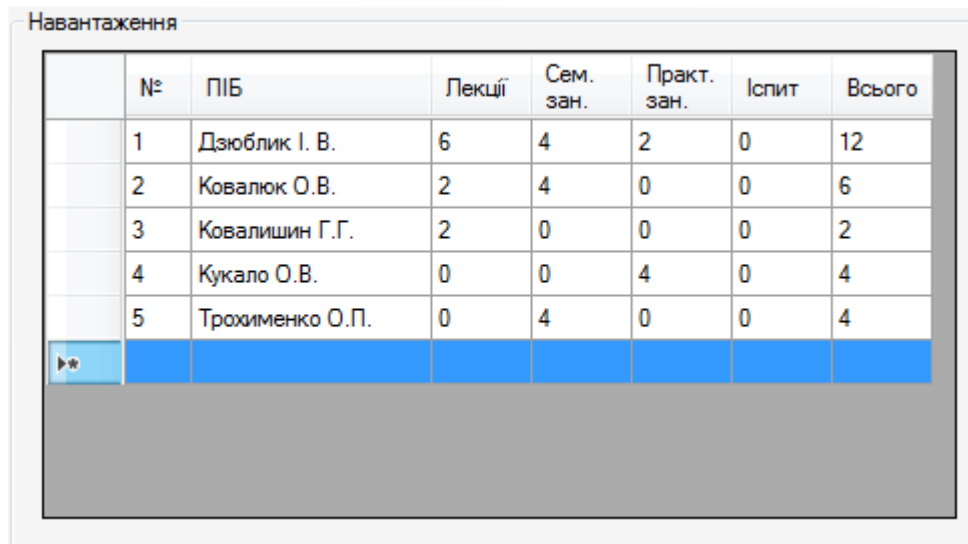
Спеціалізація: Вірусологія  
Викладач: Дзюблик І. В.  
Розрахувати

Лекції 6 годин  
Семінари 4 годин  
Практики 2 годин  
Іспит 0 годин  
Всього 12 годин

Рисунок 5.18 — розраховані дані для викладача Дзюблик І.В. за парами, які вона викладає в циклі вірусології

Ці розрахунки реалізовано за допомогою коду, який містить параметризовані SQL-запити.

Далі, щоб занести дані до таблиці, користувач має натиснути на кнопку “Додати до таблиці”. Таким чином після того, як будуть оброблені дані щодо усіх викладачів в рамках однієї спеціалізації, буде отримано DataGridView наступного вигляду (рисунок 5.19):



	№	ПІБ	Лекції	Сем. зан.	Практ. зан.	Іспит	Всього
	1	Дзюблик І. В.	6	4	2	0	12
	2	Ковалюк О.В.	2	4	0	0	6
	3	Ковалишин Г.Г.	2	0	0	0	2
	4	Кукало О.В.	0	0	4	0	4
	5	Трохименко О.П.	0	4	0	0	4

Рисунок 5.19 — приклад таблиці з навантаженням

Далі цю таблицю можна зберегти до Excel-файлу, натиснувши кнопку “Зберегти навантаження в Excel”. Ця таблиця відкриється в програмі Microsoft Excel, де користувач може внести правки вручну, якщо це необхідно, та зберегти файл.

### 5.3 Робота з програмним модулем менеджменту задач

Головне вікно “План робіт” містить:

- кнопку “Показати пріоритети виконання”, яка виділяє задачі різними кольорами відповідно до того, скільки днів залишилось до дедлайну;
- кнопку “Прибрати пріоритети виконання”, яка робить усі задачі білими;
- кнопку “Завантажити інший документ”, яка переводить користувача на другу форму, в якій він може зчитати Excel-файл;
- текстову підказку, про те, що означає кожен колір.

Вікно виглядає наступним чином (рисунок 5.20):

Номер_п_п	Рішення_вченої_ради	Дата_та_номер_протоколу	Термін_виконання	Відповідальні_особи_за_рішення	Етап_виконання	Виконано
1	Розробити (переглянути) методичні розробки семінарських та практичних занять циклів. Термін виконання: згідно з термінами робочих планів.	07.02.2018 р. протокол № 2	20.06.2019	Зав. кафедри Загорій В.А.	Методичні розробки семінарських та практичних занять циклів на стадії доопрацювання.	Виконано
2	Розробити (переглянути) тези лекцій викладачів.	07.02.2018 р. протокол № 2	01.06.2019	Зав. кафедри Загорій В.А.	Тези лекцій упорядковуються	Виконується
3	Розробити презентаційні матеріали лекцій, семінарських/практичних занять з мультимедійним супроводом.	07.02.2018 р. протокол № 2	30.04.2019	Зав. кафедри Загорій В.А.	Презентаційні матеріали лекцій розробляються та оновлюються.	Не виконано
5	Підготувати новий курс	13.04.2019	30.06.2019	Зам. декана	Підготовка, розробка матеріалів	Виконується
6	Купити підручники	7.02.2019	30.08.2019	Зав. кафедрою	Закупівля	Виконується

Зелений - до дедлайну більше 30 днів  
Жовтий - до дедлайну більше 15 днів  
Червоний - до дедлайну менше 15 днів

Рисунок 5.20 — вікно “План робіт”

Приклад виділення задач кольором (рисунок 5.21):

Номер_п_п	Рішення_вченої_ради	Дата_та_номер_протоколу	Термін_виконання	Відповідальні_особи_за_рішення	Етап_виконання	Виконано
1	Розробити (переглянути) методичні розробки семінарських та практичних занять циклів. Термін виконання: згідно з термінами робочих планів.	07.02.2018 р. протокол № 2	20.06.2019	Зав. кафедри Загорій В.А.	Методичні розробки семінарських та практичних занять циклів на стадії доопрацювання.	Виконано
2	Розробити (переглянути) тези лекцій викладачів.	07.02.2018 р. протокол № 2	01.06.2019	Зав. кафедри Загорій В.А.	Тези лекцій упорядковуються	Виконується
3	Розробити презентаційні матеріали лекцій, семінарських/практичних занять з мультимедійним супроводом.	07.02.2018 р. протокол № 2	30.04.2019	Зав. кафедри Загорій В.А.	Презентаційні матеріали лекцій розробляються та оновлюються.	Не виконано
5	Підготувати новий курс	13.04.2019	30.06.2019	Зам. декана	Підготовка, розробка матеріалів	Виконується
6	Купити підручники	7.02.2019	30.08.2019	Зав. кафедрою	Закупівля	Виконується

Показати пріоритети виконання   Прибрати пріоритети виконання   Завантажити інший документ

Рисунок 5.21 — приклад роботи кнопки “Показати пріоритети виконання”

Як було написано вище, натиснувши на кнопку “Завантажити інший документ”, користувач перейде до вікна, де він може завантажити документ Excel з побічними задачами. Важливо, щоб документ було створено з правильною структурою і саме в четвертому стовпчику містилися дати кінцевих строків виконання задач, бо програмний код зчитує інформацію саме по цьому стовпчику.

Вікно “Завантаження з Excel” в початковому стані має наступний вигляд (рисунки 5.22):

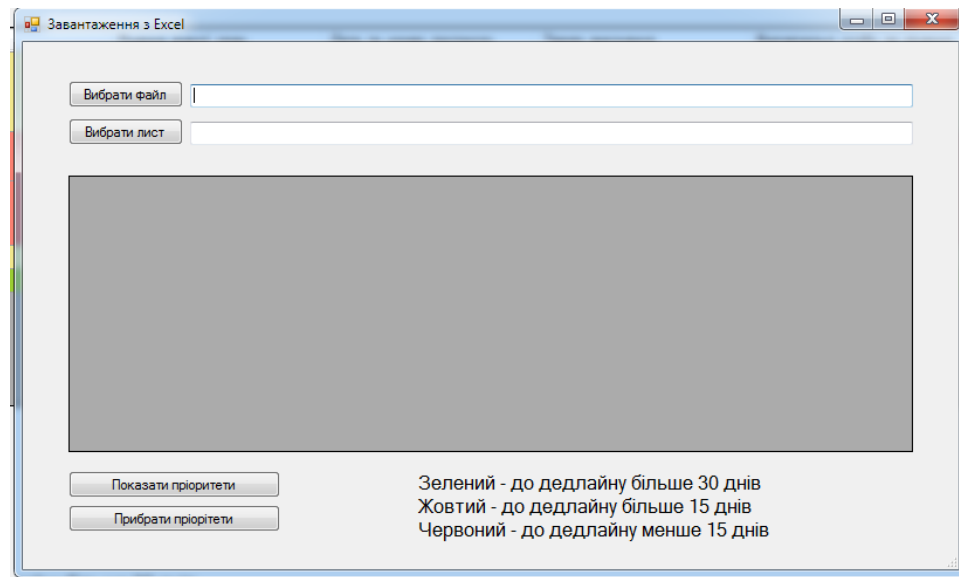


Рисунок 5.22 — вікно “Завантаження з Excel”

Коли користувач натискає кнопку “Вибрати файл”, відкривається вікно з файлами на комп’ютері, серед яких необхідно вибрати потрібний (рисунки 5.23).

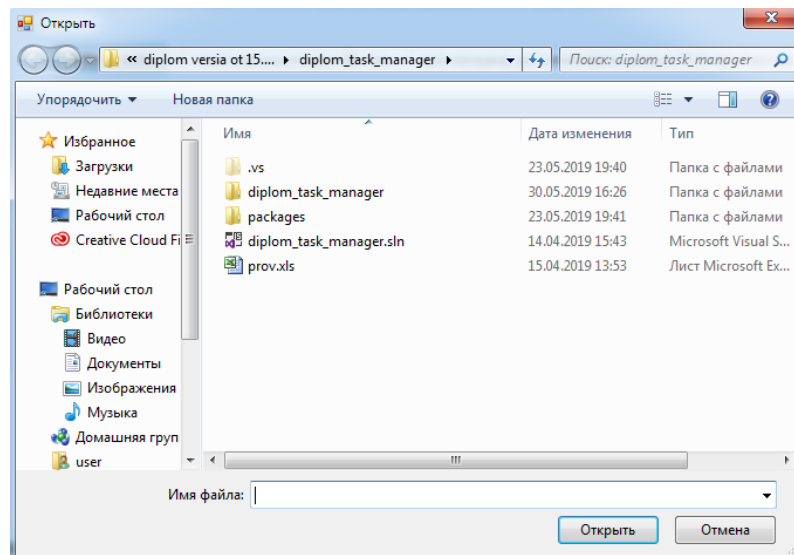


Рисунок 5.23 — вікно вибору файла

У textBox поруч з кнопкою “Вибрати лист”, необхідно написати назву листа в документі Excel, з яким користувач хоче працювати і натиснути цю кнопку.

Далі у DataGridView завантажиться таблиця з даними з обраного Excel-файла і сценарій роботи з ним такий самий, як і у вікні “План робіт” (рисунок 5.24).

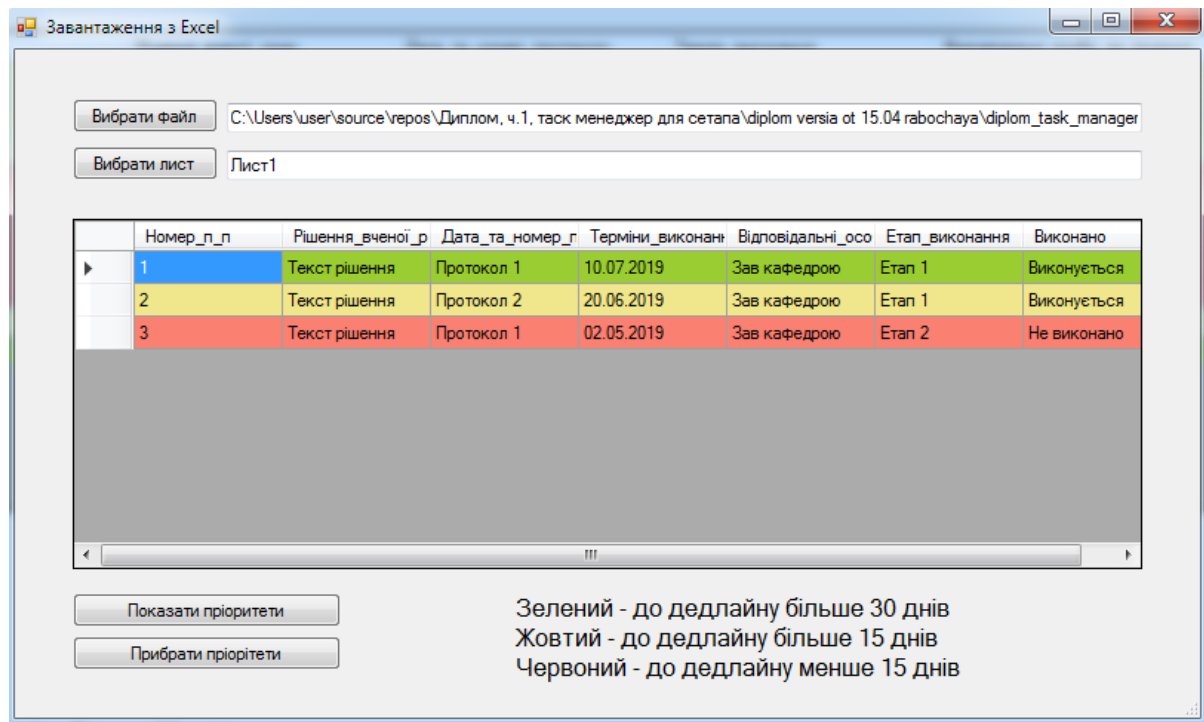


Рисунок 5.24 — робота у вікні “Завантаження з Excel”

## 5.4 Висновки до розділу 5

У розділі номер 5 було описано базові вимоги до системи для коректної роботи системи ефективного менеджменту роботи закладів післядипломної освіти та які програми мають бути встановлені попередньо.

Детально розглянуто усі можливі сценарії роботи користувача з системою та надано рекомендації по роботі з додатком.

## ВИСНОВКИ

В ході виконання даної роботи було розроблено інформаційну систему для ефективного менеджменту роботи закладів післядипломної освіти.

Додаток було написано на мові програмування C# з використанням графічного інтерфейсу Windows Forms та СУБД Microsoft SQL Server.

Програму можна використовувати для менеджменту предметів, що вивчаються в закладі, аудиторного та викладацького фонду, створення розкладу навчання та відслідковування дедлайнів плану робіт кафедри.

В ході роботи було проведено огляд та зроблено аналіз засобів, що були використані для створення даного програмного забезпечення (середовища розробки Visual Studio 2017, інтерфейсу Windows Forms, СУБД Microsoft SQL Server та відповідних бібліотек мови C#).

Для роботи з даним програмним забезпеченням необхідний лише комп'ютер середньої потужності та попередньо встановлені Microsoft SQL Server 2014 Express та Microsoft Excel.

Користувач має змогу власноруч вводити дані про групи, спеціалізації, предмети, викладачів та аудиторії. Користувач може створювати розклад навчання більш швидко та уникаючи помилок, які виникають при створенні розкладу вручну. Користувачеві надається можливість зберегти розклад для необхідної спеціалізації або викладача до Excel-файлу, а також сформувати та зберегти таблицю навантаження на викладача, відслідковувати дедлайни виконання робіт відповідальними особами.

Ця програма полегшує менеджмент роботи закладів післядипломної освіти та допомагає уникнути багатьох помилок та недоліків, які можуть виникати при кропітких задачах, які описані вище.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Виноградова Е. Б., Мудрова Е. Б. Информационное пространство рынка труда научно-преподавательских кадров вуза // Проблемы современной экономики. 2009. № 1 — 437с.
2. Сайт Time Doctor // Таск менеджер: как выбрать? [Электронный ресурс] — Режим доступа: <https://biz30.timedoctor.com/ru/task-manager/>
3. Аббакумов, А.А., Байнев, В. В., Пырякина, К.А. Алгоритм составления расписания занятий для высших учебных заведений // Огарев-online. Раздел "Технические науки". — 2015. — №20.
4. Абухания Амер Ю А . Модели, алгоритмы и программные средства обработки информации и принятия решений при составлении расписаний занятий на основе эволюционных методов: автореф. дис. на здобуття наук. ступеня канд. техн. наук / Абухания Амер Ю А – Новочеркасск, 2016. – 3 с.автореф. дисс.....канд. тех. наук. — Новочеркасск, 2016. — 3 с
5. Гладков Л.А. Генетические алгоритмы // Гладков Л.А., Курейчик В.В., Курейчик В.М. — М.: ФИЗМАТЛИТ, 2006. — 320 с.
6. Сайт Professor Web // Работа с Visual Studio [Электронный ресурс] — Режим доступа: [https://professorweb.ru/my/csharp/charp\\_theory/level2/2\\_4.php](https://professorweb.ru/my/csharp/charp_theory/level2/2_4.php)
7. Майо Д. Самоучитель Microsoft Visual Studio 2010 // Джо Майо. — СПб.: БХВ-Петербург, 2011. — 464 с.
8. Сайт METANIT.COM // Введение в Windows Forms [Электронный ресурс] — Режим доступа: <https://metanit.com/sharp/windowsforms/1.1.php>
9. Сайт Microsoft // DataGridView Class [Электронный ресурс] — Режим доступа: <https://docs.microsoft.com/en-us/dotnet/api/system.windows.forms.datagridview>
10. Сайт Microsoft // System.Data.Odbc Namespace [Электронный ресурс] — Режим доступа: <https://docs.microsoft.com/en-us/dotnet/api/system.data.odbc>



11. Сайт Microsoft // Документация по SQL Server [Электронный ресурс] — Режим доступа: <https://docs.microsoft.com/ru-ru/sql/sql-server/sql-server-technical-documentation?view=sql-server-2017&viewFallbackFrom=sql-server-2017>
12. Сайт Microsoft // Features Supported by the Editions of SQL Server 2014 [Электронный ресурс] — Режим доступа: <https://docs.microsoft.com/en-us/sql/getting-started/features-supported-by-the-editions-of-sql-server-2014>
13. Бошемин Б. Основы ADO.NET // Боб Бошемин. — М.: Вильямс, 2003. — 448 с.
14. Леоненков А. Самоучитель UML / Александров Леоненков. — СПб.: БХВ-Петербург, 2004. — 432 с.
15. Энтони М. SQL. Сборник рецептов / Молинаро Энтони // SQL. Сборник рецептов / Молинаро Энтони. — Санкт-Петербург: Символ-плюс, 2009. — С. 116.
16. Сайт Microsoft // Worksheet.Cells Property [Электронный ресурс] — Режим доступа: <https://docs.microsoft.com/en-us/dotnet/api/microsoft.office.tools.excel.worksheet.cells>
17. Сайт METANIT.COM // Работа с датами и временем [Электронный ресурс] — Режим доступа: <https://metanit.com/sharp/tutorial/19.1.php>

## ДОДАТОК А

Інформаційна система ефективного менеджменту роботи закладів  
післядипломної освіти

Специфікація

УКР.НТУУ”КПІ імені Ігоря Сікорського”\_ТЕФ\_АПЕПС\_ТР5275\_19Б

Аркушів 1

Київ 2019

Позначення	Найменування	Примітки
Документація		
УКР.НТУУ”КПІ імені Ігоря Сікорського”_ТЕФ_АПЕПС_ ТР5275_19Б	Записка.docx	Пояснювальна записка
Компоненти		
УКР.НТУУ”КПІ імені Ігоря Сікорського”_ТЕФ_АПЕПС_ ТР5275_19Б 12-1	schedule_varTest.cs Subject.cs Form1.cs	Основні компоненти
УКР.НТУУ”КПІ імені Ігоря Сікорського”_ТЕФ_АПЕПС_ ТР5275_19Б 13-1	Додаток В.doc	Опис програмного модуля

## ДОДАТОК Б

Інформаційна система ефективного менеджменту роботи закладів  
післядипломної освіти

Лістинг програми

УКР.НТУУ”КПІ імені Ігоря Сікорського”\_ТЕФ\_АПЕПС\_ТР5275\_19Б  
12-1

Аркушів 12

Київ 2019

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.Odbc;
using System.Data.SqlClient;

namespace schedule
{
    public partial class schedule_varTest : Form
    {
        OdbcConnection connectionODBC;
        string connectionStringODBC = "Driver={ODBC Driver 11 for SQL Server};server=USER-
ПК;trusted_connection=Yes;app=Microsoft® Visual Studio®;wsid=USER-ПК;database=scheduleTest";

        SqlConnection connectionSQL;
        string connectionStringSQL = "Data Source=USER-ПК;Initial
Catalog=scheduleTest;Integrated Security=True";

        public schedule_varTest()
        {
            InitializeComponent();
        }

        private void scheduleVarTestBindingNavigatorSaveItem_Click(object sender, EventArgs
e)
        {
            this.Validate();
            this.scheduleVarTestBindingSource.EndEdit();
            this.tableAdapterManager.UpdateAll(this.scheduleTestDataSet);
        }

        private void schedule_varTest_Load(object sender, EventArgs e)
        {
            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "scheduleDataSet.TypeSubj". При необходимости она может быть перемещена или удалена.
            this.typeSubjTableAdapter.Fill(this.scheduleDataSet.TypeSubj);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "scheduleDataSet.Specialization". При необходимости она может быть перемещена или удалена.
            this.specializationTableAdapter.Fill(this.scheduleDataSet.Specialization);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "scheduleDataSet.Room". При необходимости она может быть перемещена или удалена.
            this.roomTableAdapter.Fill(this.scheduleDataSet.Room);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "scheduleDataSet.Teacher". При необходимости она может быть перемещена или удалена.
            this.teacherTableAdapter.Fill(this.scheduleDataSet.Teacher);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "scheduleDataSet.Groups". При необходимости она может быть перемещена или удалена.
            this.groupsTableAdapter.Fill(this.scheduleDataSet.Groups);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "scheduleDataSet.Subject". При необходимости она может быть перемещена или удалена.
            this.subjectTableAdapter.Fill(this.scheduleDataSet.Subject);
            // TODO: данная строка кода позволяет загрузить данные в таблицу
            "scheduleTestDataSet.scheduleVarTest". При необходимости она может быть перемещена или
            удалена.
            this.scheduleVarTestTableAdapter.Fill(this.scheduleTestDataSet.scheduleVarTest);
        }
    }
}

```

```

}

public void dataReload()
{
    string command = "SELECT * from scheduleVarTest";
    using (connectionODBC = new OdbcConnection(connectionStringODBC))
    {
        if (connectionODBC.State == ConnectionState.Open)
            connectionODBC.Close();

        connectionODBC.Open();

        OdbcCommand SelectCommand = new OdbcCommand(command, connectionODBC);
        OdbcDataAdapter MyDA = new OdbcDataAdapter(SelectCommand);
        // SelectCommand.ExecuteNonQuery();

        System.Data.DataTable table = new System.Data.DataTable();
        MyDA.Fill(table);

        BindingSource bSource = new BindingSource();
        bSource.DataSource = table;

        scheduleVarTestDataGridView.DataSource = bSource;
        connectionODBC.Close();
    }
}

public void addToSchedule()
{
    System.Data.DataTable table = new System.Data.DataTable();

    OdbcParameter id;
    OdbcParameter day;
    OdbcParameter lesson;
    OdbcParameter typeLesson;
    OdbcParameter kodLesson;
    OdbcParameter themeLesson;
    OdbcParameter hours;
    OdbcParameter group;
    OdbcParameter teacher;
    OdbcParameter room;
    OdbcParameter specialization;

    try
    {
        id = new OdbcParameter();
        id.Value = int.Parse(textBox3.Text);
        day = new OdbcParameter();
        day.Value = DateTime.Parse(textBox1.Text);
        lesson = new OdbcParameter();
        lesson.Value = int.Parse(textBox2.Text);
        typeLesson = new OdbcParameter();
        typeLesson.Value = int.Parse(comboBox1.Text);
        kodLesson = new OdbcParameter();
        kodLesson.Value = comboBox2.Text;
        themeLesson = new OdbcParameter();
        themeLesson.Value = comboBox3.Text;
        hours = new OdbcParameter();
        hours.Value = int.Parse(comboBox7.Text);
        group = new OdbcParameter();
    }
}

```

```

        group.Value = comboBox4.Text;
        teacher = new OdbcParameter();
        teacher.Value = comboBox5.Text;
        room = new OdbcParameter();
        room.Value = comboBox6.Text;
        specialization = new OdbcParameter();
        specialization.Value = comboBox11.Text;

        string command = "INSERT INTO scheduleVarTest(id, Дата, Папа,
Тип_заняття_id, Код_заняття, Тематика_заняття, Кількість_годин, Група, Викладач, Аудиторія,
Спеціалізація) VALUES(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
        using (connectionODBC = new OdbcConnection(connectionStringODBC))
        {
            if (connectionODBC.State == ConnectionState.Open)
                connectionODBC.Close();

            connectionODBC.Open();

            try
            {
                OdbcCommand insertCommand = new OdbcCommand(command,
connectionODBC);

                insertCommand.Parameters.Add(id);
                insertCommand.Parameters.Add(day);
                insertCommand.Parameters.Add(lesson);
                insertCommand.Parameters.Add(typeLesson);
                insertCommand.Parameters.Add(kodLesson);
                insertCommand.Parameters.Add(themeLesson);
                insertCommand.Parameters.Add(hours);
                insertCommand.Parameters.Add(group);
                insertCommand.Parameters.Add(teacher);
                insertCommand.Parameters.Add(room);
                insertCommand.Parameters.Add(specialization);
                int res = insertCommand.ExecuteNonQuery();
                dataReload();
            }
            catch (OdbcException ex)
            {
                MessageBox.Show(ex.Message + " odbc error");
            }
            finally
            {
                connectionODBC.Close();
            }
        }
    }
    catch
    {
        MessageBox.Show("Make sure data is correct");
    }
}

public static bool button1_pressed = false;

private void Button1_Click(object sender, EventArgs e)
{
    for (int i = 0; i < scheduleVarTestDataGridView.RowCount; i++)
    {
        if (scheduleVarTestDataGridView.Rows[i].Cells[1].Value.ToString() ==
DateTime.Parse(textBox1.Text).ToString() &&
scheduleVarTestDataGridView.Rows[i].Cells[2].Value.ToString() == textBox2.Text &&
scheduleVarTestDataGridView.Rows[i].Cells[9].Value.ToString() == comboBox6.Text &&

```

```

scheduleVarTestDataGridView.Rows[i].Cells[3].Value.ToString() == "1" &&
scheduleVarTestDataGridView.Rows[i].Cells[8].Value.ToString() != comboBox5.Text)
{
    int res = i + 1;
    MessageBox.Show("На цій лекції вже є викладач. Див. рядок №" + res);
}
else if (scheduleVarTestDataGridView.Rows[i].Cells[1].Value.ToString() ==
DateTime.Parse(textBox1.Text).ToString() &&
scheduleVarTestDataGridView.Rows[i].Cells[2].Value.ToString() == textBox2.Text &&
scheduleVarTestDataGridView.Rows[i].Cells[8].Value.ToString() == comboBox5.Text &&
scheduleVarTestDataGridView.Rows[i].Cells[3].Value.ToString() != "1")
{
    int res = i + 1;
    MessageBox.Show("У викладача " + comboBox5.Text + " вже є пара в цей
час. Див. рядок №" + res);
}
else if (scheduleVarTestDataGridView.Rows[i].Cells[1].Value.ToString() ==
DateTime.Parse(textBox1.Text).ToString() &&
scheduleVarTestDataGridView.Rows[i].Cells[2].Value.ToString() == textBox2.Text &&
scheduleVarTestDataGridView.Rows[i].Cells[9].Value.ToString() == comboBox6.Text &&
scheduleVarTestDataGridView.Rows[i].Cells[3].Value.ToString() != "1")
{
    int res = i + 1;
    MessageBox.Show("Аудиторія № " + comboBox6.Text + " зайнята в цей час.
Див. рядок №" + res);
}
else if (scheduleVarTestDataGridView.Rows[i].Cells[1].Value.ToString() ==
DateTime.Parse(textBox1.Text).ToString() &&
scheduleVarTestDataGridView.Rows[i].Cells[2].Value.ToString() == textBox2.Text &&
scheduleVarTestDataGridView.Rows[i].Cells[7].Value.ToString() == comboBox4.Text &&
scheduleVarTestDataGridView.Rows[i].Cells[3].Value.ToString() != "1")
{
    int res = i + 1;
    MessageBox.Show("У групи " + comboBox4.Text + " вже є пара в цей час.
Див. рядок №" + res);
}
else if (scheduleVarTestDataGridView.Rows[i].Cells[1].Value.ToString() ==
DateTime.Parse(textBox1.Text).ToString() &&
scheduleVarTestDataGridView.Rows[i].Cells[2].Value.ToString() == textBox2.Text &&
scheduleVarTestDataGridView.Rows[i].Cells[3].Value.ToString() == "1" &&
scheduleVarTestDataGridView.Rows[i].Cells[8].Value.ToString() == comboBox5.Text &&
scheduleVarTestDataGridView.Rows[i].Cells[7].Value.ToString() == comboBox4.Text &&
scheduleVarTestDataGridView.Rows[i].Cells[9].Value.ToString() == comboBox6.Text)
{
    int res = i + 1;
    MessageBox.Show("Така лекція вже існує. Див. рядок №" + res);
}
}
button1_pressed = true;
}

private void button6_Click(object sender, EventArgs e)
{
    if (button1_pressed)
    {
        addToSchedule();
        button1_pressed = false;
    }
    else MessageBox.Show("Спочатку перевірте дані. Натисніть кнопку Перевірити");
}

private void button4_Click(object sender, EventArgs e)

```



```

{
    dataReload();
}

private void comboBox12_SelectedIndexChanged(object sender, EventArgs e)
{
    var table = new DataTable();
    SqlParameter spec = new SqlParameter();
    SqlParameter teach = new SqlParameter();
    SqlParameter subj = new SqlParameter();

    spec.ParameterName = "@spec";
    spec.Value = comboBox9.Text;
    teach.ParameterName = "@teach";
    teach.Value = comboBox10.Text;
    subj.ParameterName = "@subj";
    subj.Value = comboBox12.Text;

    string command;
    if (comboBox9.Text == "" && comboBox10.Text == "" && comboBox12.Text == "" ||
comboBox12.Text == "5")
        command = "SELECT * FROM scheduleVarTest";
    else if (comboBox9.Text == "" && comboBox10.Text != "" && comboBox12.Text != "")
        command = "SELECT * FROM scheduleVarTest WHERE Викладач = @teach and
Тип_заняття_id = @subj";
    else
        command = "SELECT * FROM scheduleVarTest WHERE Спеціалізація = @spec and
Викладач = @teach and Тип_заняття_id = @subj";

    using (connectionSQL = new SqlConnection(connectionStringSQL))
    {
        if (connectionSQL.State == ConnectionState.Open)
            connectionSQL.Close();

        connectionSQL.Open();

        try
        {
            SqlCommand selectCommand = new SqlCommand(command, connectionSQL);
            selectCommand.Parameters.Add(spec);
            selectCommand.Parameters.Add(teach);
            selectCommand.Parameters.Add(subj);
            table.Load(selectCommand.ExecuteReader());
            scheduleVarTestDataGridView.DataSource = table;
        }
        catch (SqlException ex)
        {
            MessageBox.Show(ex.Message + "sql error");
        }
        finally
        {
            connectionSQL.Close();
        }
    }
    int sum = 0;
    for (int i = 0; i < scheduleVarTestDataGridView.Rows.Count; ++i)
    {
        sum += Convert.ToInt32(scheduleVarTestDataGridView.Rows[i].Cells[6].Value);
    }
    label121.Text = sum.ToString();
}

```

```

private void comboBox9_SelectedIndexChanged(object sender, EventArgs e)
{
    var table = new System.Data.DataTable();
    OdbcParameter parameter = new OdbcParameter();

    parameter.Value = comboBox9.Text;
    // parameter.
    string command;

    if (comboBox9.Text == "")
        command = "SELECT * FROM scheduleVarTest";
    else
        command = "SELECT * FROM scheduleVarTest WHERE Спеціалізація = ?";

    using (connectionODBC = new OdbcConnection(connectionStringODBC))
    {
        if (connectionODBC.State == ConnectionState.Open)
            connectionODBC.Close();

        connectionODBC.Open();

        try
        {
            OdbcCommand selectCommand = new OdbcCommand(command, connectionODBC);
            selectCommand.Parameters.Add(parameter);
            table.Load(selectCommand.ExecuteReader());
            scheduleVarTestDataGridView.DataSource = table;
        }
        catch (OdbcException ex)
        {
            MessageBox.Show(ex.Message + "odbc error");
        }
        finally
        {
            connectionODBC.Close();
        }
    }
}

private void comboBox10_SelectedIndexChanged(object sender, EventArgs e)
{
    var table = new System.Data.DataTable();
    OdbcParameter spec = new OdbcParameter();
    spec.Value = comboBox9.Text;

    OdbcParameter teach = new OdbcParameter();
    teach.Value = comboBox10.Text;
    // parameter.
    string command;

    if (comboBox10.Text == "" && comboBox9.Text == "")
        command = "SELECT * FROM scheduleVarTest";
    else if (comboBox9.Text == "" && comboBox10.Text != "")
        command = "SELECT * FROM scheduleVarTest WHERE Викладач = ?";
    else
        command = "SELECT * FROM scheduleVarTest WHERE Викладач = ? and
Спеціалізація = ?";
    using (connectionODBC = new OdbcConnection(connectionStringODBC))
    {
        if (connectionODBC.State == ConnectionState.Open)
            connectionODBC.Close();
    }
}

```

```

        connectionODBC.Open();

        try
        {
            OdbcCommand selectCommand = new OdbcCommand(command, connectionODBC);
            selectCommand.Parameters.Add(teach);
            selectCommand.Parameters.Add(spec);
            table.Load(selectCommand.ExecuteReader());
            scheduleVarTestDataGridView.DataSource = table;
        }
        catch (OdbcException ex)
        {
            MessageBox.Show(ex.Message + "odbc error");
        }
        finally
        {
            connectionODBC.Close();
        }
    }

    int sum = 0;
    for (int i = 0; i < scheduleVarTestDataGridView.Rows.Count; ++i)
    {
        sum += Convert.ToInt32(scheduleVarTestDataGridView.Rows[i].Cells[6].Value);
    }
    label15.Text = sum.ToString();
}

private void label12_Click(object sender, EventArgs e)
{
}

private void button5_Click(object sender, EventArgs e)
{
    Microsoft.Office.Interop.Excel.Application ExcelApp = new
Microsoft.Office.Interop.Excel.Application();
    Microsoft.Office.Interop.Excel.Workbook ExcelWorkBook;
    Microsoft.Office.Interop.Excel.Worksheet ExcelWorkSheet;
    //Книга.
    ExcelWorkBook = ExcelApp.Workbooks.Add(System.Reflection.Missing.Value);
    //Таблица.
    ExcelWorkSheet =
(Microsoft.Office.Interop.Excel.Worksheet)ExcelWorkBook.Worksheets.get_Item(1);

    for (int i = 0; i < dataGridView1.Rows.Count; i++)
    {
        ExcelApp.Cells[i + 2, 1] = dataGridView1.Rows[i].Cells[0].Value;
        ExcelApp.Cells[i + 2, 2] = dataGridView1.Rows[i].Cells[1].Value;
        ExcelApp.Cells[i + 2, 3] = dataGridView1.Rows[i].Cells[2].Value;
        ExcelApp.Cells[i + 2, 4] = dataGridView1.Rows[i].Cells[3].Value;
        ExcelApp.Cells[i + 2, 5] = dataGridView1.Rows[i].Cells[4].Value;
        ExcelApp.Cells[i + 2, 6] = dataGridView1.Rows[i].Cells[5].Value;
        ExcelApp.Cells[i + 2, 7] = dataGridView1.Rows[i].Cells[6].Value;
        ExcelApp.Cells[i + 2, 8] = dataGridView1.Rows[i].Cells[7].Value;
    }

    ExcelApp.Cells[1, 1] = "№";
    ExcelApp.Cells[1, 2] = "ПИБ";
    ExcelApp.Cells[1, 3] = "Посада";
}

```

```

ExcelApp.Cells[1, 4] = "Лекції, годин";
ExcelApp.Cells[1, 5] = "Сем. зан., годин";
ExcelApp.Cells[1, 6] = "Практ. зан., годин";
ExcelApp.Cells[1, 7] = "Іспит, годин";
ExcelApp.Cells[1, 8] = "Всього";

ExcelWorkSheet.Columns[1].ColumnWidth = 20;
ExcelWorkSheet.Columns[2].ColumnWidth = 20;
ExcelWorkSheet.Columns[3].ColumnWidth = 20;
ExcelWorkSheet.Columns[4].ColumnWidth = 20;
ExcelWorkSheet.Columns[5].ColumnWidth = 20;
ExcelWorkSheet.Columns[6].ColumnWidth = 20;
ExcelWorkSheet.Columns[7].ColumnWidth = 20;
ExcelWorkSheet.Columns[8].ColumnWidth = 20;

ExcelWorkSheet.Columns[1].HorizontalAlignment =
Microsoft.Office.Interop.Excel.XlHAlign.xlHAlignLeft;
ExcelWorkSheet.Columns[2].HorizontalAlignment =
Microsoft.Office.Interop.Excel.XlHAlign.xlHAlignCenter;
ExcelWorkSheet.Columns[3].HorizontalAlignment =
Microsoft.Office.Interop.Excel.XlHAlign.xlHAlignCenter;
ExcelWorkSheet.Columns[4].HorizontalAlignment =
Microsoft.Office.Interop.Excel.XlHAlign.xlHAlignLeft;
ExcelWorkSheet.Columns[4].VerticalAlignment =
Microsoft.Office.Interop.Excel.XlVAlign.xlVAlignCenter;
ExcelWorkSheet.Columns[5].HorizontalAlignment =
Microsoft.Office.Interop.Excel.XlHAlign.xlHAlignCenter;
ExcelWorkSheet.Columns[6].HorizontalAlignment =
Microsoft.Office.Interop.Excel.XlHAlign.xlHAlignCenter;
ExcelWorkSheet.Columns[8].HorizontalAlignment =
Microsoft.Office.Interop.Excel.XlHAlign.xlHAlignLeft;

ExcelWorkSheet.Cells.Style.WrapText = true;

ExcelApp.Visible = true;
ExcelApp.UserControl = true;
}

private void button2_Click(object sender, EventArgs e)
{
    Microsoft.Office.Interop.Excel.Application ExcelApp = new
Microsoft.Office.Interop.Excel.Application();
    Microsoft.Office.Interop.Excel.Workbook ExcelWorkBook;
    Microsoft.Office.Interop.Excel.Worksheet ExcelWorkSheet;
    //Книга.
    ExcelWorkBook = ExcelApp.Workbooks.Add(System.Reflection.Missing.Value);
    //Таблиця.
    ExcelWorkSheet =
(Microsoft.Office.Interop.Excel.Worksheet)ExcelWorkBook.Worksheets.get_Item(1);

    for (int i = 0; i < scheduleVarTestDataGridView.Rows.Count; i++)
    {
        ExcelApp.Cells[i + 2, 1] =
scheduleVarTestDataGridView.Rows[i].Cells[1].Value;
        ExcelApp.Cells[i + 2, 2] =
scheduleVarTestDataGridView.Rows[i].Cells[2].Value;
        ExcelApp.Cells[i + 2, 3] =
scheduleVarTestDataGridView.Rows[i].Cells[4].Value;
    }
}

```

```

        ExcelApp.Cells[i + 2, 4] =
scheduleVarTestDataGridView.Rows[i].Cells[5].Value;
        ExcelApp.Cells[i + 2, 5] =
scheduleVarTestDataGridView.Rows[i].Cells[6].Value;
        ExcelApp.Cells[i + 2, 6] =
scheduleVarTestDataGridView.Rows[i].Cells[7].Value;
        ExcelApp.Cells[i + 2, 7] =
scheduleVarTestDataGridView.Rows[i].Cells[8].Value;
        ExcelApp.Cells[i + 2, 8] =
scheduleVarTestDataGridView.Rows[i].Cells[9].Value;

    }

    ExcelApp.Cells[1, 1] = "Дата";
    ExcelApp.Cells[1, 2] = "Пара";
    ExcelApp.Cells[1, 3] = "Код заняття";
    ExcelApp.Cells[1, 4] = "Тематика заняття";
    ExcelApp.Cells[1, 5] = "Кількість годин";
    ExcelApp.Cells[1, 6] = "Група";
    ExcelApp.Cells[1, 7] = "Викладач";
    ExcelApp.Cells[1, 8] = "Ауд.";

    ExcelWorkSheet.Columns[1].ColumnWidth = 20;
    ExcelWorkSheet.Columns[2].ColumnWidth = 20;
    ExcelWorkSheet.Columns[3].ColumnWidth = 20;
    ExcelWorkSheet.Columns[4].ColumnWidth = 20;
    ExcelWorkSheet.Columns[5].ColumnWidth = 20;
    ExcelWorkSheet.Columns[6].ColumnWidth = 20;
    ExcelWorkSheet.Columns[7].ColumnWidth = 20;
    ExcelWorkSheet.Columns[8].ColumnWidth = 20;

    ExcelWorkSheet.Columns[1].HorizontalAlignment =
Microsoft.Office.Interop.Excel.XlHAlign.xlHAlignLeft;
    ExcelWorkSheet.Columns[2].HorizontalAlignment =
Microsoft.Office.Interop.Excel.XlHAlign.xlHAlignCenter;
    ExcelWorkSheet.Columns[3].HorizontalAlignment =
Microsoft.Office.Interop.Excel.XlHAlign.xlHAlignCenter;
    ExcelWorkSheet.Columns[4].HorizontalAlignment =
Microsoft.Office.Interop.Excel.XlHAlign.xlHAlignLeft;
    ExcelWorkSheet.Columns[4].VerticalAlignment =
Microsoft.Office.Interop.Excel.XlVAlign.xlVAlignCenter;
    ExcelWorkSheet.Columns[5].HorizontalAlignment =
Microsoft.Office.Interop.Excel.XlHAlign.xlHAlignCenter;
    ExcelWorkSheet.Columns[6].HorizontalAlignment =
Microsoft.Office.Interop.Excel.XlHAlign.xlHAlignCenter;
    ExcelWorkSheet.Columns[8].HorizontalAlignment =
Microsoft.Office.Interop.Excel.XlHAlign.xlHAlignLeft;

    ExcelWorkSheet.Cells.Style.WrapText = true;

    ExcelApp.Visible = true;
    ExcelApp.UserControl = true;
}

private void button7_Click(object sender, EventArgs e)
{
    foreach (DataGridViewRow row in scheduleVarTestDataGridView.Rows)
    {
        int days = DateTime.Parse(row.Cells[1].Value.ToString()).Day - 1;
        if (days % 2 == 0)

```

```

        row.DefaultCellStyle.BackColor = System.Drawing.Color.LightGray;
    else
        row.DefaultCellStyle.BackColor = System.Drawing.Color.White;
    }
}

private void comboBox5_SelectedIndexChanged(object sender, EventArgs e)
{
}

private void label15_Click(object sender, EventArgs e)
{
}

private void button3_Click(object sender, EventArgs e)
{
    var table = new System.Data.DataTable();
    SqlParameter spec = new SqlParameter();
    spec.ParameterName = "@spec";
    spec.Value = comboBox13.Text;

    SqlParameter teach = new SqlParameter();
    teach.ParameterName = "@teach";
    teach.Value = comboBox14.Text;

    SqlParameter spec1 = new SqlParameter();
    spec1.ParameterName = "@spec1";
    spec1.Value = comboBox13.Text;

    SqlParameter teach1 = new SqlParameter();
    teach1.ParameterName = "@teach1";
    teach1.Value = comboBox14.Text;

    SqlParameter spec2 = new SqlParameter();
    spec2.ParameterName = "@spec2";
    spec2.Value = comboBox13.Text;

    SqlParameter teach2 = new SqlParameter();
    teach2.ParameterName = "@teach2";
    teach2.Value = comboBox14.Text;

    SqlParameter spec3 = new SqlParameter();
    spec3.ParameterName = "@spec3";
    spec3.Value = comboBox13.Text;

    SqlParameter teach3 = new SqlParameter();
    teach3.ParameterName = "@teach3";
    teach3.Value = comboBox14.Text;

    SqlParameter spec4 = new SqlParameter();
    spec4.ParameterName = "@spec4";
    spec4.Value = comboBox13.Text;

    SqlParameter teach4 = new SqlParameter();
    teach4.ParameterName = "@teach4";
    teach4.Value = comboBox14.Text;
}

```

```

// parameter.
string command;
string command_lek;
string command_sem;
string command_pr;
string command_exam;

command = "SELECT SUM(Кількість_годин) FROM scheduleVarTest WHERE Викладач =
@teach and Спеціалізація = @spec";
command_lek = "SELECT SUM(Кількість_годин) FROM scheduleVarTest WHERE Викладач =
@teach1 and Спеціалізація = @spec1 and Тип_заняття_id = 1";
command_sem = "SELECT SUM(Кількість_годин) FROM scheduleVarTest WHERE Викладач =
@teach2 and Спеціалізація = @spec2 and Тип_заняття_id = 2";
command_pr = "SELECT SUM(Кількість_годин) FROM scheduleVarTest WHERE Викладач =
@teach3 and Спеціалізація = @spec3 and Тип_заняття_id = 3";
command_exam = "SELECT SUM(Кількість_годин) FROM scheduleVarTest WHERE Викладач
= @teach4 and Спеціалізація = @spec4 and Тип_заняття_id = 4";
using (connectionSQL = new SqlConnection(connectionStringSQL))
{
    if (connectionSQL.State == ConnectionState.Open)
        connectionSQL.Close();

    connectionSQL.Open();

    try
    {
        SqlCommand selectCommand = new SqlCommand(command, connectionSQL);
        selectCommand.Parameters.Add(teach);
        selectCommand.Parameters.Add(spec);
        object result4 = selectCommand.ExecuteScalar();
        var res4 = result4 == DBNull.Value ? 0 : Convert.ToInt32(result4);
        label33.Text = res4.ToString();

        SqlCommand selectCommandLek = new SqlCommand(command_lek,
connectionSQL);
        selectCommandLek.Parameters.Add(teach1);
        selectCommandLek.Parameters.Add(spec1);
        object result3 = selectCommandLek.ExecuteScalar();
        var res3 = result3 == DBNull.Value ? 0 : Convert.ToInt32(result3);
        label29.Text = res3.ToString();

        SqlCommand selectCommandSem = new SqlCommand(command_sem,
connectionSQL);
        selectCommandSem.Parameters.Add(teach2);
        selectCommandSem.Parameters.Add(spec2);
        object result2 = selectCommandSem.ExecuteScalar();
        var res2 = result2 == DBNull.Value ? 0 : Convert.ToInt32(result2);
        label30.Text = res2.ToString();

        SqlCommand selectCommandPr = new SqlCommand(command_pr, connectionSQL);
        selectCommandPr.Parameters.Add(teach3);
        selectCommandPr.Parameters.Add(spec3);
        object result1 = selectCommandPr.ExecuteScalar();
        var res1 = result1 == DBNull.Value ? 0 : Convert.ToInt32(result1);
        label31.Text = res1.ToString();

        SqlCommand selectCommandExam = new SqlCommand(command_exam,
connectionSQL);
        selectCommandExam.Parameters.Add(teach4);
        selectCommandExam.Parameters.Add(spec4);
        object result = selectCommandExam.ExecuteScalar();
        var res = result == DBNull.Value ? 0 : Convert.ToInt32(result);

```

```

        label32.Text = res.ToString();
    }
    catch (SQLException ex)
    {
        MessageBox.Show(ex.Message + "odbc error");
    }
    finally
    {
        connectionSQL.Close();
    }
}

private void button8_Click(object sender, EventArgs e)
{
    dataGridView1.Rows.Add(" ", comboBox14.Text, label29.Text, label30.Text,
label31.Text, label32.Text, label33.Text);
}
}
}

```



## ДОДАТОК В

Інформаційна система ефективного менеджменту роботи закладів  
післядипломної освіти

Опис програмного модулю

УКР.НТУУ”КПІ імені Ігоря Сікорського”\_ТЕФ\_АПЕПС\_ТР5275\_19Б

13-1

Аркушів 8

Київ 2019

## АНОТАЦІЯ

Розділ містить опис частини, яка слугує для роботи з базою даних для модуля формування розкладу, що є структурною одиницею програмного продукту. Модуль надає можливість заносити нові дані, оновлювати та видаляти наявні. Перевіряти введені дані, формувати розклад, сортувати його за певними критеріями та зберігати до Excel. Модуль написано мовою програмування C#, з використанням технології Microsoft SQL Server, ADO.NET та бібліотек Microsoft.Office.Interop.Excel, System.Data.SqlClient та System.Data.Odbc.

## ЗМІСТ

ЗАГАЛЬНІ ВІДОМОСТІ .....	3
ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ .....	4
ОПИС ЛОГІЧНОЇ СТРУКТУРИ .....	5
ВИКОРИСТОВУВАНІ ТЕХНІЧНІ ЗАСОБИ.....	6
ВИКЛИК І ЗАВАНТАЖЕННЯ .....	7
ВХІДНІ І ВИХІДНІ ДАНІ .....	8

## ЗАГАЛЬНІ ВІДОМОСТІ

У додатку розглянуто модуль формування розкладу закладу післядипломної освіти з кодом УКР.НТУУ”КПІ”\_ТЕФ\_АПЕПС\_TV41107\_18Б 12-1, що міститься у файлі `schedule_varTest.cs`. Модуль реалізовано завдяки провайдерам даних `SqlClient` та `Odbc`, а також технології `ADO.NET`. Модуль призначений для управління формування розкладу. Користувач може заносити нові дані до розкладу, попередньо їх перевіряючи, видаляти та редагувати наявні записи, сортувати таблицю з розкладом, автоматично формувати таблицю навантажень на викладачів.

## **ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ**

Модуль створено для складання коректного розкладу. Дані, що вводяться користувачем, проходять усі необхідні перевірки. Після того, як сформована повна таблиця розкладу, в користувача є можливість сформулювати таблицю навантажень та викладачів та відсортувати записи з розкладу, потрібні для конкретного випадку, та зберегти ці дані в Excel.

## ОПИС ЛОГІЧНОЇ СТРУКТУРИ

При запуску інформаційної системи користувач бачить початкове вікно програми, з якого може перейти до інших вікон, де може вводити відповідні дані, передивлятися їх, коригувати або видаляти. У вікні “Оформлення розкладу користувач” також може сортувати дані за різними критеріями та зберігати розклад та навантаження на викладачів у Excel-файли.

## **ВИКОРИСТОВУВАНІ ТЕХНІЧНІ ЗАСОБИ**

Середовищем розробки програмного модуля було обрано Microsoft Visual Studio 2017 та засоби бібліотек Microsoft.Office.Interop.Excel, System.Data.SqlClient та System.Data.Odbc із застосуванням технології ADO.NET.

Системою управління базами даних було обрано Microsoft SQL Server 2014 Express.

## **ВИКЛИК І ЗАВАНТАЖЕННЯ**

Розроблений модуль не потребує інсталяції, треба запустити проект “schedule.sln” в Microsoft Visual Studio 2017 або запустити файл “schedule.exe”.

Після запуску програми користувач переходить до початкового меню програми, звідки може перейти на необхідну форму та виконувати потрібні дії.



## **ВХІДНІ І ВИХІДНІ ДАНІ**

Вхідними даними для модуля є дані, які користувач може вводити в базу даних за допомогою Microsoft SQL Server або вводити в програмному продукті.

Вихідними даними програмного модуля є Excel-файли з інформацією, яку користувач обирає у додатку.

## ДОДАТОК Г

Інформаційна система ефективного менеджменту роботи закладів  
післядипломної освіти

Апробації

УКР.НТУУ”КПІ імені Ігоря Сікорського”\_ТЕФ\_АПЕПС\_ТР5275\_19Б

Аркушів 3

Київ 2019

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

# СУЧАСНІ ПРОБЛЕМИ НАУКОВОГО ЗАБЕЗПЕЧЕННЯ ЕНЕРГЕТИКИ

Матеріали XVII Міжнародної  
науково-практичної конференції  
молодих вчених та студентів  
м. Київ, 23-26 квітня 2019 року,

ТОМ 2



Київ - 2019

<b>Природні ресурси регіону. Система обліку лісових ресурсів.</b>	164
<i>ЛІНЬОВ Д.О., студент гр. ТМ-51</i>	
<i>Керівник - ст.викл. Шульженко О.Ф.</i>	
<b>Інтернет-система редагування та аналізу генетичних послідовностей.</b>	165
<i>ЖИРОВ М.І., студент гр. ТІ-51</i>	
<i>Керівник - ст.викл. Бандурка О.І.</i>	
<b>Google maps API. Обробка даних отриманих з пристрою та їх відображення на карті веб-інтерфейсу.</b>	166
<i>ЖИРНОВ А.Ю., студент гр. ТВ-51</i>	
<i>Керівник - доц., к.т.н. Смаковський Д.С.</i>	
<b>Природні ресурси регіону. Система обліку водних ресурсів.</b>	167
<i>ДАВИДЧУК О.С., студент гр. ТМ-51</i>	
<i>Керівник - ст.викл. Шульженко О.Ф.</i>	
<b>Інформаційна система біоінформаційного та епідеміологічного аналізу вакцинопрофілактики гострих вірусних інфекцій.</b>	168
<i>ГОРБАТЮК О.І., студент гр. ТР-52</i>	
<i>Керівник - ст.викл. Бандурка О.І.</i>	
<b>Алгоритмізація та реалізація алгоритму аналізу інформації щодо визначення та прогнозу статусу гравця/команди гравців.</b>	169
<i>ГОНЧАР О.В., студент гр. ТР-52</i>	
<i>Керівник - доц., к.т.н. Коваль О.В.</i>	
<b>Природні ресурси регіону. Система обліку лікувальних ресурсів.</b>	170
<i>ГАРКУША О.В., студент гр. ТМ-51</i>	
<i>Керівник - ст.викл. Шульженко О.Ф.</i>	
<b>Система управління процесом утворення водню з відновлюваної целюлозовмісної сировини у біореакторі.</b>	171
<i>БОНДАРЕНКО А.В., студент гр. ТІ-51</i>	
<i>Керівник - ст.викл. Бандурка О.І.</i>	
<b>Автоматизована система для моделювання та аналізу адміністративних функцій організацій.</b>	172
<i>БИТИК М.О., студент гр. ТМ-51</i>	
<i>Керівник - доц., к.т.н. Кузьмініх В.О.</i>	
<b>Інформаційна система ефективного менеджменту роботи закладів післядипломної освіти.</b>	173
<i>БАСАЛИК Д.А., студент гр. ТР-52</i>	
<i>Керівник - ст.викл. Бандурка О.І.</i>	
<b>Інформаційна система аналізу профілактики хронічних вірусних інфекцій.</b>	174
<i>БАСАЛИК Г.А., студент гр. ТР-52</i>	
<i>Керівник - ст.викл. Бандурка О.І.</i>	
<b>Розробка сайту для продажу продуктів здорового харчування.</b>	175
<i>СЕМІОН В.І., студент гр. ТМ-62</i>	
<i>Керівник - ст.викл., Шульженко О.Ф.</i>	
<b>Організація по впровадженню та обслуговуванню автоматизованого обладнання теплиць.</b>	176
<i>ЛОКОТАРЬОВ Є.О., студент гр. ТМ-62;</i>	
<i>ШАПОВАЛ В.О., студент гр. ТМ-62</i>	
<i>Керівник - ст.викл. Шульженко О.Ф.</i>	
<b>Розробка прикладного програмного забезпечення для автоматизації роботи кінологічного клубу.</b>	177
<i>ЛЕБЕДИК Т.О., студент гр. ТМ-62;</i>	
<i>САПОН О.М., студент гр. ТМ-62</i>	

**УДК 621.43.056:632.15**

Студент 4 курсу, гр. ТР-52 Басалик Д.А.  
Ст.викл. Бандурка О.І.

## **ІНФОРМАЦІЙНА СИСТЕМА ЕФЕКТИВНОГО МЕНЕДЖМЕНТУ РОБОТИ ЗАКЛАДІВ ПІСЛЯДИПЛОМНОЇ ОСВІТИ**

В сучасному світі перед керівниками навчальних закладів все частіше постає питання автоматизації певних процесів. Зокрема, автоматизація створення графіку навчального процесу та менеджменту задач, які повинні бути виконані співробітниками закладів[1]. Перевагами таких систем є економія часу співробітників, а також структурованість та організованість процесу виконання[2]. Неможливо забути про якість завдання або дедлайн, тому що вони відображаються в програмі.

Основна проблема існуючих додатків – це ресурсозатратність через наявність зайвого функціоналу, який не використовується, та відсутність враховування специфіки закладів освіти. Через перенавантаженість інтерфейсу також не завжди очевидно, за що відповідає кожен елемент програми, і витрачається багато часу на те, щоб зрозуміти, як цим користуватися[3].

Програмне забезпечення буде складатися з двох модулів, які є окремими технічними рішеннями. Це допоможе на кожній структурній одиниці вищого навчального закладу впровадити саме той модуль, який їй потрібен, або ж обидва модуля.

Основні модулі:

- Модуль менеджменту задач на основі Windows Forms – інтерфейсу для розробки візуально гармонійного додатку та елемента DataGridView – для роботи з базою даних та формування таблиць на основі цих даних[4].
- Модуль генератора розкладу навчального закладу на основі Windows Forms та DataGridView.

Програмний продукт має бути простим у використанні, інтуїтивно зрозумілим та візуально привабливим. Програма має надавати можливість рандомної генерації розкладу навчання з певними обмеженнями та можливістю подальшого редагування, а також трекер задач зі статусом виконання та нагадуваннями про дедлайн.

Програмне забезпечення вирішить проблему зірваних дедлайнів, помилок, які можливі при складанні розкладу вручну, а також збереже час співробітників закладу, завдяки автоматизації цих кропітких задач.

Перелік посилань:

1. Виноградова Е. Б., Мудрова Е. Б. Информационное пространство рынка труда научно-преподавательских кадров вуза // Проблемы современной экономики. 2009. № 1 – 437с.
2. Сайт Time Doctor // Таск менеджер: как выбрать? [Електронний ресурс] – Режим доступу: <https://biz30.timedoctor.com/ru/task-manager/>
3. Сайт Texterra // Разделяй и властвуй: 14 бесплатных сервисов для управления маркетинговыми проектами [Електронний ресурс] – Режим доступу: <https://texterra.ru/blog/razdelyay-i-vlastvuy-15-besplatnykh-servisov-dlya-upravleniya-marketingovymi-proektami.html>
4. Сайт Microsoft // DataGridView Class. [Електронний ресурс] - Режим доступу: <https://docs.microsoft.com/en-us/dotnet/api/system.windows.forms.datagridview>